

**Konzeption und Implementierung einer  
Verkehrsinformationsapplikation für Smartphones  
am Beispiel des iPhones**

**Masterarbeit**

Zur Erlangung des Akademischen  
Grades Master of Engineering “M. Eng.”

**Technische Hochschule Wildau**

**Fachbereich Ingenieurwesen  
Studiengang Telematik**

vorgelegt von:	<b>Timo Jonathan Schmidt</b>
Reg.-Nr.:	<b>TM09/13/SS2011</b>
Betreuer:	<b>Prof. Dr. rer. nat. Janett Mohnke, Prof. Dr. Bernd Eylert</b>
Themenstellender Betrieb:	<b>DLR e.V.</b>
Betrieblicher Betreuer:	<b>Dipl. Ing. Alexander Sohr</b>

---

# Bibliographische Beschreibung und Referat

**Konzeption und Implementierung einer Verkehrsinformationsapplikation für Smartphones am Beispiel des iPhones.** Masterarbeit, Technische Hochschule Wildau 2011, 141 Seiten, 63 Abbildungen, 93 Literaturangaben, 1 Beilage.

## Ziel

Konzeption und Implementierung einer App für das iPhone, welche Floating Car Data (FCD) sammelt und überträgt sowie den Anwendern durch eine leichte Bedienbarkeit und Bereitstellung interessanter, aktueller Verkehrs- und Routeninformationen einen Mehrwert bietet.

## Inhalt

In dieser Arbeit wird eine mobile Anwendung hinsichtlich der Faktoren für ein positives Nutzererlebnis (User Experience) konzipiert und implementiert. Es werden wesentliche Grundlagen über User Experience, Smartphone, Ortung, FCD und plattformübergreifende Entwicklung genannt. Die Umsetzung für das iPhone erfolgt mit dem plattformübergreifenden Framework Titanium des Unternehmens Appcelerator, um eine zukünftige Portierung für Android zu ermöglichen. Die Applikation offeriert ein grafisches Abbild der Verkehrslage, verkehrsoptimiertes Routing von einem Start- zum Zielpunkt und optionalen Zwischenpunkt, eine Routen-Überwachung sowie Offline-Karten für Berlin. Des Weiteren wird eine Ortung, je nach Vorhandensein mit GPS, Mobilfunk oder WLAN, durchgeführt.

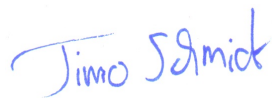
# Abstract

This master thesis is about a mobile application which is designed and implemented to ensure a positive user experience. It presents basic considerations in relation to the user experience, Smartphones, mobile phone positioning, Floating Car Data (FCD) and cross-platform development. The iPhone implementation is based on the cross-platform framework Titanium supplied by Appcelerator, to enable the application to be easily ported to Android in the future. The app provides a Visualisation of the traffic situation, optimized routing from the starting point to the destination with an optional intermediate point, route monitoring and offline maps of Berlin. Additionally, the current position of the mobile phone is determined by GPS, the mobile phone network or a Wi-Fi network.

# Selbständigkeitserklärung

Hiermit versichere ich, die vorliegende Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Wildau, den 17. November 2011



Timo Jonathan Schmidt

# Abkürzungsverzeichnis

<b>A-GPS</b>	Assisted Global Positioning System
<b>API</b>	Application Programming Interface
<b>BDSG</b>	Bundesdatenschutzgesetz
<b>BPSK</b>	Bipolar-Phase Shift Key
<b>C/A</b>	Coarse/Acquisition
<b>CD</b>	Compact Disc
<b>CSS</b>	Cascading Stylesheets
<b>DLR</b>	Deutsches Zentrum für Luft- und Raumfahrt
<b>FCD</b>	Floating Car Data
<b>GPS</b>	Global Positioning System
<b>GSM</b>	Global System for Mobile Communications
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IEEE</b>	The Institute of Electrical and Electronics Engineers
<b>JDBC</b>	Java Database Connectivity
<b>KML</b>	Keyhole Markup Language
<b>LOI</b>	Location of Interest
<b>OS</b>	Operating System
<b>OSM</b>	OpenStreet Map
<b>PDA</b>	Personal Digital Assistant
<b>SDK</b>	Software Development Kit
<b>SOAP</b>	Simple Object Access Protocol
<b>StVO</b>	Straßenverkehrs-Ordnung

**SQL** Structured Query Language

**TOA** Time Of Arrival

**US** United States

**UTC** Universal Time Coordinated

**WiFi/Wi-Fi** Wireless Fidelity (Synonym für WLAN in Amerika)

**WSDL** Web Service Description Language

**WLAN** Wireless Local Area Network

**XML** Extensible Markup Language

# Glossar

**Open-Source** Sammlung von Lizenzen, bei denen die Software folgende rechtliche Grundlage besitzt: Frei verfügbare und offengelegte Software, die modifiziert werden kann.

**Prototyp** Ein Vor-Exemplar/Versuchsprodukt zum Erproben und Verifizieren von Funktionen. Es werden Eigenschaften untersucht, um später aus dem Prototyp ein Produkt für den Vertrieb zu formen. In Bezug auf die Softwareentwicklung kann ein Prototyp auch als Alpha-Version bezeichnet werden.

**Verkehrslage** Aktuelle Verkehrssituation in einem bestimmten Gebiet. In dieser Arbeit wird die grafische Darstellung der Verkehrslage auf dem Straßennetz beschrieben. Üblicherweise werden Straßenabschnitte bei Stau mit Rot, stockenden (langsamer Verkehr) Zuständen mit Gelb und fließendem Verkehr mit Grün dargestellt.

**Webview** Ein natives Element (native Klasse), in dem Webinhalt dargestellt wird. Durch dieses Konstrukt können in einer nativen mobilen Applikation externe oder interne Websites angezeigt werden.



# Inhaltsverzeichnis

<b>Bibliographische Beschreibung und Referat</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Selbstständigkeitserklärung</b>	<b>iii</b>
<b>Abkürzungsverzeichnis</b>	<b>iv</b>
<b>Glossar</b>	<b>vi</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Gegenstand und Motivation . . . . .	1
1.2. Anforderungen und Zielsetzung . . . . .	3
1.3. Gliederung der Arbeit . . . . .	4
1.4. Abgrenzung der Arbeit . . . . .	4
<b>2. Fundamentale Grundlagen und Analysen</b>	<b>5</b>
2.1. Das Smartphone . . . . .	5
2.1.1. Entwicklung des Smartphone-Marktes und Faktoren für Kundenzufriedenheit . . . . .	6
2.1.2. Die mobile Applikation - App . . . . .	8
2.2. User Experience und Apple's iOS Richtlinien . . . . .	9
2.2.1. Das Human Interface für Human Computer Interaction . . . . .	10
2.2.2. iOS Charakteristiken . . . . .	12
2.3. Native und plattformübergreifende Entwicklung einer App . . . . .	17
2.3.1. Hybride Frameworks . . . . .	18
2.4. Verkehrsdatenquellen und Informationsanbieter . . . . .	24

2.4.1.	Verkehrsinformationen des DLR . . . . .	24
2.4.2.	Floating Car Data (FCD) . . . . .	25
2.5.	Positionsbestimmung eines Smartphones . . . . .	31
2.5.1.	Positionsermittlung durch das Global Positioning System (GPS) . . . . .	31
2.5.2.	Mobilfunkortung . . . . .	34
2.5.3.	Wi-Fi . . . . .	37
2.5.4.	Details zur Ortung bei Apple Produkten . . . . .	38
2.5.5.	Alternative Ortungsmöglichkeiten . . . . .	40
<b>3.</b>	<b>Systematische Analysen</b>	<b>41</b>
3.1.	Ist- und Sollzustand . . . . .	41
3.1.1.	Istzustand . . . . .	41
3.1.2.	Sollzustand . . . . .	46
3.2.	Anwendungsfallanalyse (Use Case) . . . . .	47
3.2.1.	Aus Sicht des Nutzers . . . . .	47
3.2.2.	Aus Sicht des Unternehmens . . . . .	47
3.3.	Alternative Verkehrsinformationsapplikationen für Smartphones . . . . .	49
3.3.1.	Google . . . . .	49
3.3.2.	Navigon . . . . .	49
3.4.	Anforderungsanalyse . . . . .	51
3.4.1.	Anforderungsmatrix . . . . .	51
3.5.	Analyse der Bewertungsmatrix . . . . .	53
3.5.1.	Auswahl des Lösungsweges . . . . .	53
<b>4.</b>	<b>Konzept und Umsetzung der mobilen Applikation</b>	<b>55</b>
4.1.	Konzept . . . . .	55
4.1.1.	Brauchbarkeit und Nutzbarkeit . . . . .	56
4.1.2.	Interaktion . . . . .	64
4.1.3.	Barrierefreiheit . . . . .	65
4.1.4.	Anleitung . . . . .	65
4.1.5.	Design . . . . .	66
4.1.6.	Benutzerschnittstelle . . . . .	67
4.2.	Installation und Einstieg . . . . .	69
4.2.1.	Entwicklungsumgebung . . . . .	70

4.3.	Umsetzung des Konzeptes und Implementierung . . . . .	71
4.3.1.	Tabbar-Elemente . . . . .	71
4.3.2.	Globale Events . . . . .	72
4.3.3.	Systemtypische Buttons und Navigationsbar . . . . .	73
4.3.4.	Kommunikation mit dem Server . . . . .	73
4.3.5.	Die Karte als Webview . . . . .	74
4.3.6.	Ortung und FCD-Handhabung . . . . .	77
<b>5.</b>	<b>Umsetzung der serverseitigen Komponenten</b>	<b>80</b>
5.1.	Datenmodell . . . . .	80
5.1.1.	Beschreibung der Datenbankstrukturen . . . . .	81
5.1.2.	Umsetzung . . . . .	82
5.2.	Web Service . . . . .	85
5.2.1.	Web Service Funktionalitäten für die App . . . . .	86
5.2.2.	Routing-Engine und Routen-Monitoring . . . . .	88
5.2.3.	OpenStreet Map Offline-Karten . . . . .	90
<b>6.</b>	<b>Testphase und Identifizierung von Qualitätsmerkmalen</b>	<b>93</b>
6.1.	Überprüfung von der FCD-Erhebung . . . . .	94
6.1.1.	Ursachenanalyse und Verbesserungsansatz . . . . .	97
6.2.	Untersuchung der Nutzerakzeptanz . . . . .	98
6.2.1.	Fragenkatalog . . . . .	98
6.3.	Plattformübergreifende Entwicklung . . . . .	100
<b>7.</b>	<b>Ergebnisse und Ausblick</b>	<b>102</b>
7.1.	Beschreibung und Bewertung der Vorgehensweise . . . . .	102
7.2.	Zusammenfassung der Ergebnisse . . . . .	102
7.2.1.	Die App in Bildern . . . . .	103
7.3.	Ausblick . . . . .	105
7.4.	Persönliches Fazit . . . . .	106
<b>8.</b>	<b>Zusammenfassung</b>	<b>108</b>
<b>Anhang</b>		<b>109</b>
A.	Informationen über die Beilage (Compact Disc) . . . . .	109

B.	Weitere Informationen zum Thema GPS . . . . .	110
B.1.	Das Global Positioning System (GPS) . . . . .	110
B.2.	GPS Status Meldung vom 17.07.2011 . . . . .	123
C.	Human Interface und User Experience . . . . .	124
D.	Testphase . . . . .	125
D.1.	Fragenkatalog . . . . .	125
D.2.	Batch Skript für die automatische Erzeugung einer KML-Datei	126
<b>Abbildungsverzeichnis</b>		<b>127</b>
<b>Literaturverzeichnis</b>		<b>130</b>

---

# 1. Einleitung

## 1.1. Gegenstand und Motivation

In der heutigen Zeit hat die Mobilität sowohl in der Wirtschaft als auch in der Gesellschaft einen hohen Stellenwert. Die signifikante Relevanz einer effizienten Fortbewegung initiiert kontinuierlich viel Betrieb im Straßenverkehr durch den öffentlichen Personennah- und motorisierten Individual- und Güterverkehr. Analog vermehren sich Unfälle und Verkehrsstörungen, die zu einer Verzögerung und somit zu einer unvorteilhaften Mobilität führen. Hierdurch entsteht beim Institut für Verkehrssystemtechnik des Deutschen Zentrums für Luft- und Raumfahrt e.V. (DLR), Standort Berlin, die beständige Motivation durch wissenschaftliche Arbeiten eine Erhöhung der Sicherheit und Effizienz des Straßenverkehrs zu insistieren und zu erzielen. Substanziell ist die Erfassung von Verkehrsdaten, um eine Stauung oder Stockung<sup>1</sup> auf den Straßen zu erkennen und gegebenenfalls umfahren zu können. Für die Forschung werden im Institut Verkehrsinformationen in der Art von Floating Car Data (FCD) gesammelt, verarbeitet und analysiert. Die FCD-Informationen beinhalten im Verkehrsfluss von Fahrzeugen erhobene Daten, wie die Position und den Zeitpunkt. Anhand dieser Informationen werden die Reisezeiten und -geschwindigkeiten für die zutreffenden Straßenabschnitte berechnet. Die Fahrzeuge fungieren als Systemteilnehmer und verfügen über eine Hardware-Ausrüstung, um im Global Positioning System (GPS) Daten empfangen und diese über das mobile Internet übertragen zu können<sup>2</sup>. Als Informationsquellen wirken Kooperationspartner, wie zum Beispiel die Taxizentrale Berlin, die gegen Entgelt Taxidaten bereitstellt. Ungefähr 4300 Taxifahrer bilden für das DLR in Berlin einen umfangreichen Datenbestand, denn

---

<sup>1</sup>In Kapitel 3.1.1 wird eine Unterscheidung und Wertigkeit dieser Verkehrszuständen (Stauung und Stockung) nahegelegt.

<sup>2</sup>Weitere Informationen zu dem FCD-System folgen im Kapitel 2.4.2.

## 1.1. Gegenstand und Motivation

---

die Taxizentrale erhebt von aktiven Taxifahrern kontinuierlich Informationen und leitet diese an das Institut weiter. Im Gegenzug befähigen die analysierten und verarbeiteten Daten das DLR - beispielsweise der Verkehrsmanagement-Zentrale (VMZ) Berlin - aussagekräftige Verkehrsinformationen gegen Entgelt zur Verfügung zu stellen. Die Generierung von Verkehrs- und Reisezeitinformationen aus FCD-Verkehrsquellen basiert auf dem über mehrere Jahre aufgebauten Know-how des Unternehmens. Diese Technologie befindet sich beim DLR bereits seit 10 Jahren in der Anwendung. Kontinuierlich wird die Technik weiter erforscht und entwickelt [Brockfeld 24.05.2011]. In Bezug auf die Gesamtheit einer größeren Menge kooperierender Fahrzeuge in einem bestimmten Territorium spricht man von einer "Fahrzeugflotte". Für Stadtteile und Straßen können Aussagen über den Verkehrszustand getroffen werden, wenn die Fahrzeuge der Flotte diese befahren. Das DLR verfügt nicht nur in Berlin, sondern zum Beispiel ebenfalls in München und Nürnberg<sup>3</sup>, über eine Kooperation und somit in verschiedenen urbanen Räumen über einen hohen Datenbestand durch Taxifлотten. Die Resultate ermöglichen dem Unternehmen, die Verkehrslagedaten der Städte übergreifend zu evaluieren. Hieraus folgt, dass das Institut nur in Abhängigkeit von Kooperationspartnern in bestimmten Gebieten Verkehrslagen durch FCD erhebt. Demzufolge ist das FCD-System bisher in zweifacher Sachlage begrenzt. Zum einen können Aussagen über Verkehrszustände nur an Stellen getroffen werden, an denen in Großstädten das Territorium von Partnern durchfahren wird. Zum anderen ist die Qualität der Prognosen an den prozentualen Ausstattungsgrad und das Intervall von Sendungen gebunden. Der prozentuale Ausstattungsgrad wird durch die Anzahl von Bezugsquellen der Partner am System im Verhältnis zu allen Verkehrsteilnehmern gebildet. Auf die fixierten Intervalle von Sendungen hat das DLR keinen Einfluss, da die zeitlichen Abstände von der Taxizentrale festgelegt werden. Die Genauigkeit der Verkehrsprognosen und die Ausbreitung der territorialen Abdeckung erweitern sich mit zunehmender Quantität von akquirierten FCD-Quellen.

Diese Masterarbeit wird einen bedeutungsvollen Teil zur Weiterentwicklung und Verbesserung des fundierten FCD-Systems beitragen. Die folgende Fragestellung bildet die Initiative und den Anlass dieser Arbeit: Wie können möglichst viele Privatpersonen erreicht und als Teilnehmer am System gewonnen werden? Als Beant-

---

<sup>3</sup>Alle aktuell unterstützten Städte werden in Kapitel 2.4.2 aufgezeigt.

## 1.2. Anforderungen und Zielsetzung

---

wortung erweist es sich, zugunsten einer prävalenten Smartphone-Plattform, eine mobile Applikation zu kreieren, welche FCD an das DLR sendet. Die signifikante Herausforderung besteht darin, eine Applikation zu entwickeln, die innovativ, nutzerfreundlich und attraktiv erscheint und zudem einen Mehrwert aufzeigt. Zielgerichtet nimmt sich die Masterarbeit dieser Herausforderung an. Im Rahmen der Arbeit wird für das Smartphone "iPhone" von Apple [Apple Inc. 2011b] eine mobile Applikation konzeptionell entworfen und prototypisch umgesetzt.

## 1.2. Anforderungen und Zielsetzung

Ziel der Arbeit ist die Schaffung einer mobilen Anwendung, die sowohl einen vernehmbaren Mehrwert bietet, als auch für Nutzer leicht bedienbar und anwendbar ist. Es wird eine Applikation unter dem Namen "Cityrouter" für das iPhone konzeptionell entworfen und umgesetzt. Die Applikation ermöglicht neue FCD-Bezugsquellen, indem die Ortungsinformationen des Smartphone-Nutzers kontinuierlich an eine bestehende Anwendung des DLR gesendet werden. Ebenso besteht die Aufgabe, die Verkehrslage von Berlin in der mobilen Applikation zur Verfügung zu stellen. Dafür wird ein grafisches Abbild der Verkehrslage in dem bereits bestehenden und verwendeten webbasierten Kartenmaterial OpenStreet Map (OSM)<sup>4</sup> des DLR dargestellt. Des Weiteren wird eine kontinuierliche Ortung des Smartphone-Nutzers durchgeführt und die lokalisierte Position in der Karte angezeigt. Darüber hinaus wird eine kurze Analyse des Smartphone-Marktes durchgeführt und die verbreitetsten Plattformen hinsichtlich ihrer weltweiten Entwicklungen durch Verkaufszahlen aufgezeigt. Hierdurch wird die rasante Entwicklung von Android, des Smartphone-Betriebssystems von Google<sup>5</sup>, ersichtlich. Um zukünftig möglichst viele Applikationsnutzer zu akquirieren, wird die Bedeutsamkeit einer potenziellen Portierung für Android deutlich. Eine weitere Aufgabe ist die Analyse bestehender Anforderungen der mobilen Applikation sowie die Identifizierung von Komponenten und Umsetzungsmöglichkeiten. Das Konzept wird für mobile Anwendung gezielt danach ausgerichtet, eine Benutzerfreundlichkeit und ein positives Nutzererlebnis zu schaffen.

---

<sup>4</sup>Internetpräsenz des Anbieters: <http://openstreetmap.de> (Zugriff: August 2011)

<sup>5</sup>Link zu der Android Plattform von Google: <http://www.google.com/mobile/android/> (Zugriff: Juli 2011)

## 1.3. Gliederung der Arbeit

Die Arbeit ist thematisch in die Abschnitte Grundlagen, systematische Analysen, Konzepte und Implementierung, Testphase sowie Ergebnisse und Ausblick gegliedert. Das Kapitel 2 umfasst die relevanten Recherchen und Informationen für diese Arbeit. Mithilfe dieses Basiswissens und anhand bestehender Anforderungen werden in Kapitel 3 systematische Analysen durchgeführt. Hier werden Anwendungsfälle, Ist- und Sollzustand wie auch die genauen Anforderungen analysiert. Die Konzepte und Implementierungen für eine mobile Applikation werden in Kapitel 4 und serverseitige Komponenten in Kapitel 5 beschrieben. Eine Dokumentation und Erläuterung der Testphase erfolgt in Kapitel 6. Das Kapitel 7 beinhaltet die Ergebnisse, einen Ausblick und persönliches Fazit.

## 1.4. Abgrenzung der Arbeit

In dieser Arbeit wird ein Einblick in die aktuelle Konstellation des Smartphone-Marktes gegeben, eine Anwendungsfallanalyse durchgeführt und wesentliche Aspekte genannt, die eine Nutzerzufriedenheit und -freundlichkeit für eine Applikation ermöglichen. Jedoch zählen ein Geschäftsmodell, die Bestimmung der Zielgruppe, das Marketing oder ähnliche betriebswirtschaftliche Aspekte nicht zu den Aufgaben dieser Arbeit. Die Lokalisierung des Smartphones wird je nach Vorhandensein mit GPS, Mobilfunk oder Wireless Local Area Network (WLAN)<sup>6</sup> durchgeführt. Diese drei Technologien sind in dem Kapitel 2.5 hinsichtlich der Fähigkeit beschrieben, wie eine Ortung durchgeführt werden kann. Weitere Informationen zu diesen Technologien sind nicht Inhalt dieser Arbeit. Die Mobilfunkortung wird anhand des Mobilfunkstandards Global System for Mobile Communications (GSM) erläutert. Weiterhin werden für den interessierten Leser - als Zusatz der eigentlichen Aufgabe - im Anhang B weitere Informationen über GPS zur Verfügung gestellt. Wie bereits erwähnt, beinhalten FCD im Verkehrsfluss von beteiligten Fahrzeugen erhobene geografische Positionen. Bei dieser Arbeit soll der optimale Fall angenommen werden, dass Applikationsnutzer entsprechende Verkehrsteilnehmer sind. Aufgrund dessen werden die gesammelten und übertragenen Daten als FCD bezeichnet.

---

<sup>6</sup>Gleichbedeutend mit Wireless Fidelity (Wi-Fi); Synonym für WLAN in Amerika [Eylert 2005, S. 70]



---

## 2. Fundamentale Grundlagen und Analysen

### 2.1. Das Smartphone

Das Smartphone, zu deutsch das "intelligente Telefon", kann als eine Vereinigung der Funktionalitäten und des Leistungsumfangs eines Mobiltelefons und PDAs<sup>1</sup> aufgefasst werden [Lin und Ye 16–17. Mai 2009, S. 617]. Ein charakterisierendes Merkmal ist das Betriebssystem (eng. Operating System (OS)) eines Smartphones. Das OS zeichnet ein intelligentes Mobiltelefon aus, denn es umfasst die Kernbestandteile, wie Applikationen, Performance und Sicherheit. Ein Smartphone erlaubt es dem Nutzer, sich zusätzliche Programme, die sogenannten Apps (siehe Kapitel 2.1.2), zu installieren und somit den Funktionsumfang des Betriebssystems individuell zu erweitern. Hierbei wird der Anwender meistens durch entsprechende Schnittstellen, beispielsweise beim iPhone von Apple durch den App Store<sup>2</sup> oder bei dem Betriebssystem Android von Google durch den Android Market<sup>3</sup>, unterstützt, indem das Auffinden und Abbonieren von verfügbaren Programmen erleichtert wird. Die zurzeit global verbreitetsten OS der Smartphones sind laut Gartner Inc. [Petty und Stevens 19. Mai 2011]:

- **Android** - Betriebssystem von Google
- **iOS** - Betriebssystem von Apple

---

<sup>1</sup>PDA (Personal Digital Assistant) ist ein handliches elektronisches Endgerät, das über Funktionen eines Minicomputers und üblicherweise über ein Adressbuch, Aufgabenplanung und Kalender verfügt. [Zhang 17–19. Nov. 2010, S.1299]

<sup>2</sup>Internetpräsenz des App Stores: <http://www.apple.com/de/iphone/apps-for-iphone> (Zugriff: August 2011)

<sup>3</sup>Link zum Android Market: <https://market.android.com> (Zugriff: August 2011)

## 2.1. Das Smartphone

---

- **Symbian** - Betriebssystem von Nokia<sup>4</sup>
- **BlackBerry** - Betriebssystem von RIM (Research in Motion)<sup>5</sup>
- **Windows** - Betriebssystem von Microsoft<sup>6</sup>

### 2.1.1. Entwicklung des Smartphone-Marktes und Faktoren für Kundenzufriedenheit

Im Februar 2011 gaben Microsoft und Nokia eine Kooperation bekannt, somit setzt Nokia in Zukunft auf das Betriebssystem Windows für Smartphone Geräte [Nokia 11. Februar 2011 – London]. Beide Unternehmen erhoffen durch diese Partnerschaft, sich im Markt besser positionieren zu können [Müller 11. Februar 2011].

<b>Company</b>	<b>1Q11 Units</b>	<b>1Q11 Market Share (%)</b>	<b>1Q10 Units</b>	<b>1Q10 Market Share (%)</b>
Android	36,267.8	36.0	5,226.6	9.6
Symbian	27,598.5	27.4	24,067.7	44.2
iOS	16,883.2	16.8	8,359.7	15.3
Research In Motion	13,004.0	12.9	10,752.5	19.7
Microsoft	3,658.7	3.6	3,696.2	6.8
Other OS	3,357.2	3.3	2,402.9	4.4
<b>Total</b>	<b>100,769.3</b>	<b>100.0</b>	<b>54,505.5</b>	<b>100.0</b>

**Abbildung 2.1.:** Weltweiter Smartphone-Verkauf anhand des Betriebssystems laut Gartner Inc. - 1. Quartal 2010 und 1. Quartal 2011  
Quelle: [Pettey und Stevens 19. Mai 2011]

In Abbildung 2.1 werden die Ergebnisse einer Studie zu dem weltweiten Smartphone-Verkauf anhand des Betriebssystems durch Gartner Inc. vom 19. Mai 2011 gezeigt. Es wird deutlich, dass Android Geräte innerhalb eines Jahres (vom 1. Quartal 2010 zum 1. Quartal 2011) mit einem Wachstum von 26,4% das von Nokia verwendete Betriebssystem Symbian von der marktführenden Position im Verkauf verdrängen

<sup>4</sup>Website des Anbieters: <http://symbian.nokia.com/> (Zugriff: August 2011)

<sup>5</sup>Link zum Unternehmen: <http://de.blackberry.com/> (Zugriff: August 2011)

<sup>6</sup>Internetpräsenz des Anbieters: <http://www.microsoft.com/windowsphone/> (Zugriff: August 2011)

## 2.1. Das Smartphone

---

konnten. Laut einer weiteren Studie vom 7. April 2011 seitens Gartner Inc. wird global in Zukunft der Anteil der verkauften Symbian Geräte stetig geringer, bis die Marktanteile 2015 im Verkauf prognostiziert bei 0,1% liegen [Petty und Stevens 7. April 2011]. Die weltweite Studie sagt voraus, dass iOS bis 2014 die zweithöchsten Marktanteile im Verkauf mit etwa einem Viertel halten würde, ein Android Gerät hingegen zu dieser Zeit jedes zweite verkaufte Smartphone sei. Des Weiteren besagt die Studie, dass die Kooperation zwischen Nokia und Microsoft sich auszahlt, denn 2012 sollen bereits 10,8% der verkauften Smartphones ein Microsoft Betriebssystem besitzen. Bis 2015 sollen sich diese Anteile sogar auf 19,8% steigern, sodass in diesem Jahr iOS durch die von Gartner Inc. vorhergesagten 17,2% vom zweithöchsten Verkaufsanteil im Markt durch Microsoft verdrängt werden würde.

Im Oktober 2010 wurde laut einer Studie von Bitkom [Bitkom 11. Februar 2011]<sup>7</sup> festgestellt, dass in Deutschland jeder fünfte Handynutzer ein Smartphone besitzt. Des Weiteren ergab in Deutschland eine Befragung von 1012 Smartphone-Nutzern durch das Marktforschungsinstitut Goldmedia Custom Research im Mai 2011, dass 29% der Befragten im Besitz eines iPhones sind, 21% eines Samsung<sup>8</sup> Gerätes<sup>9</sup>, 19% Nokia, 13% HTC<sup>10</sup> und nur 7% BlackBerry von RIM [Kerkau Mai 2011]. Diese Studie zeigt weiterhin die Zufriedenheit der Besitzer mit dem Gerät und die genauen Faktoren auf. Die wesentlichen Ergebnisse in diesem Zusammenhang werden in Abbildung 2.2 gezeigt. Hieraus geht hervor, dass bei den iPhone-Besitzern gefolgt von HTC-Nutzern die größte Zufriedenheit vorliegt und eine gute Bedienbarkeit als wichtigstes Kriterium genannt wird. Weiterhin besagt die Analyse, dass die Gesamtzufriedenheit an zweiter Stelle von der Qualität des App-Stores und dicht gefolgt vom Design des Gerätes beeinflusst wird.

---

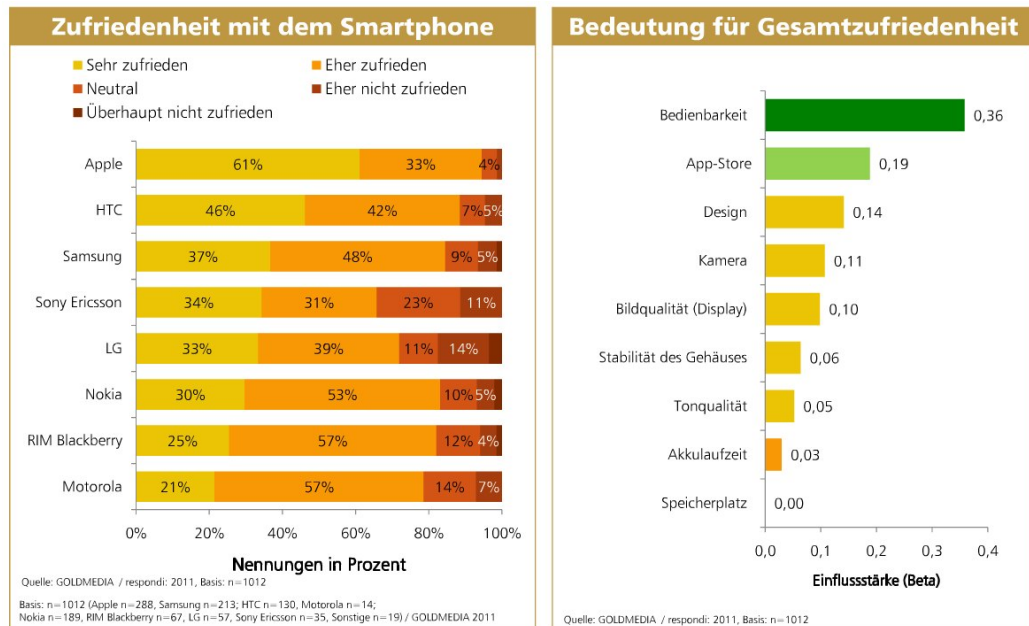
<sup>7</sup>Es wurden 849 Handynutzer ab 14 Jahren per Telefon von Forsa (Gesellschaft für Sozialforschung und statistische Analyse) befragt. Dabei konnten 22 Prozent nicht urteilen, ob sie im Besitz eines Handys oder Smartphones sind.

<sup>8</sup>Südkoreanisches Unternehmen, Website des Anbieters: <http://samsung.com> mit u.A. einer deutschen Tochtergesellschaft in Schwalbach <http://www.samsung.de/> (Zugriff: August 2011)

<sup>9</sup>Es wird in dieser Studie nicht das genaue Betriebssystem unterschieden. Samsung vertrieb bis Oktober 2010 auch Smartphones mit Symbian, jetzt nur noch Bada, Android, Windows.

<sup>10</sup>High Tech Computer Corporation, ein Hersteller von Smartphones und Tablets mit Hauptsitz in Taiwan (<http://www.htc.com/>) stellt Geräte sowohl mit Android als auch mit Windows her.

## 2.1. Das Smartphone



**Abbildung 2.2.: Smartphone Zufriedenheit**  
Quelle: [Kerkau Mai 2011]

### 2.1.2. Die mobile Applikation - App

Die im Allgemeinen verwendete Kurzbezeichnung App kann als Abkürzung für eine mobile Applikation, beziehungsweise im Englischen "Application", interpretiert werden. Unter einer App versteht man ein Programm, das ein Smartphone über systemeigene Funktionen und Applikationen hinaus erweitern kann. Es wird dabei zwischen einer nativen App und einer Web-App unterschieden, auf deren genaue Eigenschaften und Vor- und Nachteile im Kapitel 2.3 eingegangen wird.

## 2.2. User Experience und Apple's iOS Richtlinien

Der Ausdruck und die Bedeutung der User Experience werden anhand einer Konversation zwischen Donald Norman<sup>11</sup> und Peter Merholz<sup>12</sup> erläutert, die am 13. Dezember 2007 veröffentlicht wurde. Im Anhang C werden eine Aufzeichnung in Form einer Audio-Datei und die Quelle zu diesem Dialog zur Verfügung gestellt. Des Weiteren werden in dem Anhang die im Folgenden zitierten prägnanten Stellen mit entsprechenden zeitlichen Positionen im Medium und weitere Informationen zu den Zitaten nahegelegt.

(1) *"I invented the term because I thought human interface and usability were too narrow. I wanted to cover all aspects of the person's experience with the system including industrial design, graphics, the interface, the physical interaction and the manual."* Donald Norman

Laut Donald Norman (siehe Zitat (2)) ist im Zusammenhang mit der Nutzung und Erläuterung von User Experience Vorsicht geboten, da der Begriff oft im veränderten Kontext verwendet wird. Somit wird User Experience teilweise durch die zeitlich bedingte Evolution als Term in seinem eigentlichen Gedanken missinterpretiert.

(2) *"The problem with the evolution in user experience is, that it has lost its meaning."* Donald Norman

Letztendlich meint die Kreation einer User Experience die Schaffung eines positiven Nutzererlebnisses. Dieses Ereignis wird durch - wie in der Abbildung 2.3 gezeigten Abstraktion - verschiedene Prinzipien gebildet. In den nachfolgenden Kapiteln wird detailliert auf die einzelnen Komponenten, die ein attraktives und angenehmes Nutzererlebnis schaffen, eingegangen. Denn nach Angaben des Unternehmens Apple [Apple Inc. 23. März 2011, S. 9] schätzen und bemerken es Nutzer vor allem, wenn derartige Richtlinien eingehalten werden und die Applikationen den Eindruck erwecken, spezifisch für das Gerät entworfen worden zu sein:

---

<sup>11</sup>Schriftsteller, ehemaliger Professor für Kognitionswissenschaften der University of California, San Diego und für Informatik an der Northwestern University, ehemals Vizepräsident der Advanced Technology Group bei Apple

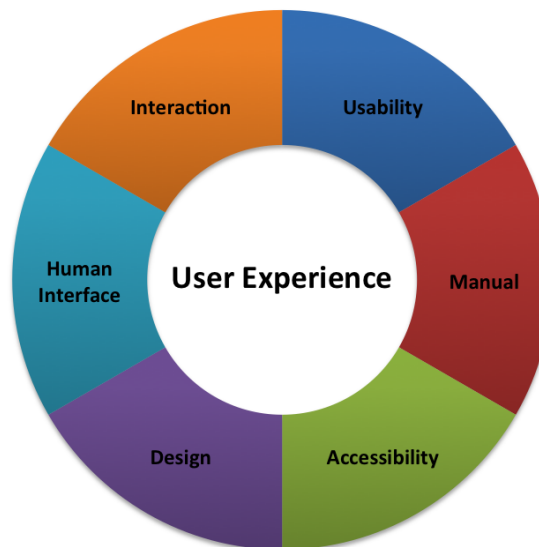
<sup>12</sup>Mitgründer von Adaptive Path, Internetpräsenz des Unternehmens: <http://adaptivepath.com/> (Zugriff: August 2011)

## 2.2. User Experience und Apple's iOS Richtlinien

---

*"People appreciate iOS apps that feel as though they were designed expressly for the device. For example, when an app fits well on the device screen and responds to the gestures that people know, it provides much of the experience people are looking for."* Apple Inc.

Es gilt, den Richtlinien zu folgen und Elemente zu identifizieren, die iOS einzigartig machen. So kann durch Detail eine positive User Experience impliziert werden.



**Abbildung 2.3.:** Abstraktion der User Experience

In Anlehnung an der Beschreibung von Donald Norman sowie Modifizierung von [Deng und Li 12–14 Dez. 2008] und [Song und Sun 22–25 Nov. 2008]

### 2.2.1. Das Human Interface für Human Computer Interaction

Das Human Interface oder Human-Machine Interface, zu deutsch die "Benutzerschnittstelle", umfasst all die Komponenten, die einem Menschen die Interaktion mit einem Gerät ermöglichen. Um komfortable und gut handhabbare Programme zu erstellen, ist eine Berücksichtigung der Ergonomie wichtig. Die Bedienung einer Anwendung durch einen Menschen kann als ideal angesehen werden, wenn das Programm intuitiv durchführbar ist. Wie in Kapitel 2.1.1 bereits erwähnt wurde, ist für eine hohe Gesamtzufriedenheit von Smartphone-Besitzern die gute Bedienbarkeit

## 2.2. User Experience und Apple's iOS Richtlinien

---

ausschlaggebend. Hieraus ist die Motivation von Apple gegründet, eine möglichst hohe Qualität bei Applikationen von Dritt-Anbietern im App Store zu erzielen. Damit Dritt-Anbieter ihr fertiges Produkt in dem App Store anbieten können, ist die Veröffentlichung bei Apple zu beantragen. Die Publizierung der App wird nur bei Erfüllung der Richtlinien und Anforderungen inszeniert. Um Entwickler im Verlauf der Konstruktion von iOS Programmen zu unterstützen, wurden die sogenannten Human Interface und Design Norm von Apple veröffentlicht [Apple Inc. 23. März 2011]. Diese Dokumentationen zeigen nicht nur einen Leitfaden und die Richtlinien für die Entwicklung von hochwertigen Programmen auf, sondern auch Empfehlungen, um die Wahrscheinlichkeit einer Zulassung in den App Store, die Benutzerfreundlichkeit und die Erkennbarkeit des Mehrwerts zu erhöhen. Die Anwender von Apps sollen nicht viel Zeit auf Hilfsseiten und beim Lesen von Anleitungen investieren müssen, bevor der Mehrwert und die Funktionsweise der Applikation erkannt werden. Hierfür legen die Dokumentationen wesentliche Techniken und Informationen für Entwickler von Apps nahe.

### Prinzipien des Human Interface beim iOS

Für einen möglichst intuitiven Umgang des Nutzers mit dem Programm ist es zum einen essenziell, gezielt Wiedererkennungselemente einzusetzen. Zum anderen kann die Intuition durch die Einhaltung von der Norm in Größen (Höhe und Weite) und Anordnungen (Koordinaten) der Komponenten gefördert werden. Die Wiedererkennungselemente sind prägnante Strukturen von Anwendungen der mobilen Plattform, die dem Nutzer somit als symptomatische Bestandteile aus anderen Applikationen bereits bekannt sein können. Zu diesen Elementen zählen zum Beispiel: der Aufbau einer grafischen Oberfläche, die typischen Navigationselemente und deren grafische Darstellung oder Anordnung. Laut Apple [Apple Inc. 23. März 2011, S. 21-23] sind für ein Human Interface im Wesentlichen folgende Prinzipien entscheidend:

- *Aesthetic Integrity* - Das Erscheinungsbild und die Darstellung der App soll zum Anwendungsbereich passen und den Funktionsumfang möglichst gut integrieren.
- *Consistency* - Verwendung von typischen UI-Controlls, -Icons, -Views und -Standards sowie diese in einem richtigen Kontext einzusetzen.

## 2.2. User Experience und Apple's iOS Richtlinien

---

- *Direct Manipulation* - Für Interaktionen gilt es, Multi-Touch Gestiken zu unterstützen. Somit fördert man eine höhere Affinität und Sensibilität für die Handhabung und Steuerung der App.
- *Feedback* - Ein Nutzer wird bei funktionalen Veränderungen oder beim Laden des Inhalts der App entsprechend informiert. Des Weiteren werden Statusmeldungen erwartet, wenn nach einer Aktion keine sofortige Reaktion der Anwendung möglich ist (z.B. Ladebalken).
- *Metaphors* - Virtuelle Grafiken, Objekte oder Aktionen werden als Metapher der realen Welt nachempfunden. Anwender können hierdurch umgehend den Nutzen und die Bedienung identifizieren.
- *User Control* - Wichtig ist eine gute Balance zwischen programmtechnischer Automatisierung und manueller Kontrolle. Der Nutzer soll möglichst eigenständig die App steuern und über Fehler informiert werden.

### 2.2.2. iOS Charakteristiken

Besonders charakteristisch für iOS ist das Display und dessen Anzeige, da es sowohl interessante Grafiken und Content zur Verfügung stellt, als auch den Anwender mit dem Gerät über Multi-Touch interagieren lässt. Die Erfahrung und der bereits bekannte Umgang eines Nutzers mit anderen Apps können bei Verwendung von Charakteristiken vereinigt werden und den Lern- und Verständnisprozess unterstützen. Einige iOS Geräte (iPhone, iPad) verfügen über unterschiedliche Auflösungen und ermöglichen zwei verschiedene Ansichten: Landscape und Portrait<sup>13</sup>. Je nachdem, ob eine App die Ansichten unterstützt, wird die Darstellung beim Drehen des Gerätes um 90° automatisch gewechselt. Für das iPhone 4 werden die Dimensionen 640 x 960 Pixel [Apple Inc. 2011b] für Portrait und 960 x 640 Pixel für Landscape angegeben. Des Weiteren werden folgende Richtlinien definiert, die für jede Auflösung zutreffend sind [Apple Inc. 23. März 2011, S. 13]:

- **Auswählbare UI Elemente sollten die Größe von 22 x 22 Pixel nicht unterschreiten.**
- **Die hohe Qualität von grafischen Darstellungen ist offensichtlich.**

---

<sup>13</sup>Landscape und Portrait sind gleichbedeutend mit Quer- und Hochformat.



## 2.2. User Experience und Apple's iOS Richtlinien

- Der hauptsächliche Fokus eines Nutzers liegt beim Content.

Die Anwender eines iPhones starten Programme über den "Home-Display"<sup>14</sup>, der sich voreingestellt im Portrait Modus befindet. Deshalb ist es sinnvoll, eine App standardmäßig ebenso im Portrait Modus zu starten, damit der Inhalt von Anfang an in der gleichen Orientierung zur Verfügung gestellt wird.

### iOS UI-Elemente

Einige prägnante iOS UI-Elemente werden in Abbildung 2.4 mittels der verwendeten Elemente in der Cityrouter App verdeutlicht und deren Bedeutung beschrieben.


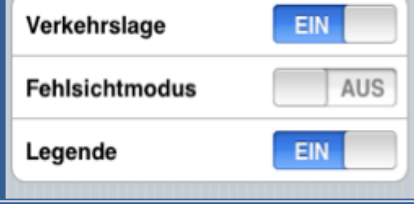
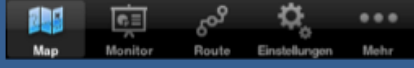
iOS Darstellung UI Element	Name und Beschreibung
	Die <b>Navigationsbar</b> befindet sich an oberster Stelle in der App. Die Navigationsbar beinhaltet wesentliche Funktionen zum Steuern und den Title. In der Cityrouter App wird der Refresh Button als rechter Navigationsbar-Button platziert. Kartenmaterial soll nur bei manueller Betätigung neugeladen werden, da neugeladene Kartendaten Traffic verursachen.
	In iOS ist der graue Hintergrund mit leicht helleren senkrechten Streifen typisch. Bezeichnungen (eng. Labels) werden als statischer Text mit einem dunkleren Grauton in den Hintergrund gezeichnet. Tabellen sind durch abgerundete Ecken markant. Eine <b>Switch</b> repräsentiert zwei Zustände, beziehungsweise den Status von zwei Einstellungsmöglichkeiten (hier EIN/AUS)
	Die <b>Tabbar</b> gibt Anwendern die Möglichkeit, zwischen verschiedenen Fenster zu wechseln.

Abbildung 2.4.: Typische iOS UI-Elemente anhand der Cityrouter App

### Applikationsfenster in iOS

Eine iOS Applikation besitzt ein Fenster, in dem der gesamte Inhalt und Funktionsumfang repräsentiert wird. Dabei erwartet der Nutzer eine sequenzielle Ausführung

<sup>14</sup>Der Hauptbildschirm eines iOS Gerätes, hier befinden sich alle Applikationen auswählbar als Symbole. Vergleichbar mit dem Desktop eines Computers.

## 2.2. User Experience und Apple's iOS Richtlinien

---

von bestimmten Sektionen in der App [Apple Inc. 23. März 2011, S. 17]. Am Beispiel der App Store Applikation von Apple, eine Eigenaufnahme wird in Abbildung 2.5 gezeigt, lässt sich dieses Prinzip veranschaulicht nachvollziehen. Die verschiedenen Sektionen von Inhalt und Funktionen werden in unterschiedlichen Tabs in einer Tabbar zur Verfügung gestellt. Hierdurch wird aber eine nacheinander folgende Bedienung des Angebotsportfolios in verschiedenen ansteuerbaren Fenstersektionen in einem Fensterrahmen der mobilen Anwendung deutlich.



Abbildung 2.5.: Screenshot des Apple App Stores (08.08.2011)

### Multi-Touch

Der wichtigste ergonomische Aspekt ist die Unterstützung und Handhabung des Multi-Touch-Displays. Bei iOS geschieht die Steuerung durch bestimmte Fingerbewegungen auf dem Display, die als Gestiken bezeichnet werden [Apple Inc. 23. März 2011, S. 14]. Für die Konzeption einer Applikation müssen diese Steuerungsmöglichkeiten gefördert werden. Um dem Anwender eine möglichst gute Ergonomie zu bieten, sollen Gestiken und deren Bedeutungen konsequent bei Applikationen eingehalten und übernommen werden. Derartige Regelmäßigkeiten sind bereits durch Programme des Betriebssystems insistiert. Es werden folgende Methodiken von iOS unterstützt [Apple Inc. 23. März 2011, S. 14]:

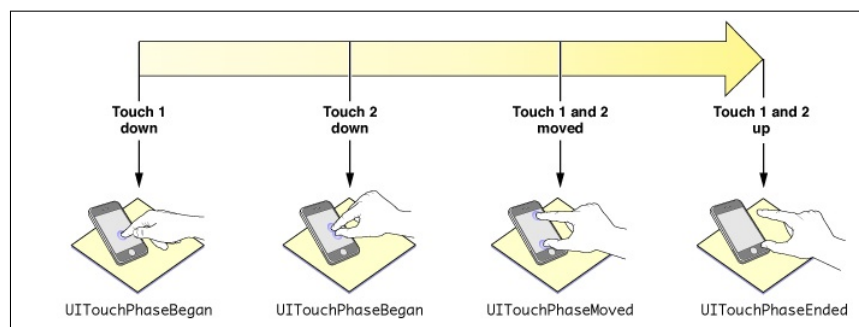
- *Tab* - Berühren des Displays zur Auswahl von Elementen.

## 2.2. User Experience und Apple's iOS Richtlinien

---

- *Pinch open and close* - Hinein- und Herauszoomen, beispielsweise auf einer Karte oder Content. Zwei Finger sind dabei auf dem Display so zu bewegen, wie bei einem Zukneifen (*Pinch close*) oder in entgegengesetzter Bewegungsrichtung wieder zu öffnen (*Pinch open*).
- *Drag* - Element auswählen, das bewegt werden kann (beispielsweise Symbole auf dem Home-Display).
- *Flick* - Display anschnippen oder -streichen, um verschiedene Inhalte von rechts nach links oder von oben nach unten zu navigieren.
- *Swipe* - Bildschirm von rechts nach links oder andersherum streichen, um beispielsweise bei Tabelleninhalten einen Löschen-Button sichtbar zu machen.
- *Double Tap* - Hineinzoomen und einen Inhalt zentrieren oder herauszoomen, falls schon ein Hineinzoom erfolgte.
- *Touch and hold* - Display berühren und Position halten, um in Texten ein Wort zu markieren oder ein Element für *Drag* auszuwählen.
- *Shake* - Gerät schütteln, um eine Auswahl oder Aktion zu annullieren.

Anhand des genannten Umfangs der interaktiven Möglichkeiten wird deutlich, dass die einzelnen Bedienungen in Sequenzen und bestimmten Phasen von Gestiken zu unterscheiden sind, um sie eindeutig zu identifizieren. In Abbildung 2.6 wird die Erkennung von einer *Pinch and open* Interaktion veranschaulicht.



**Abbildung 2.6.:** Sequenz und Phasen von Multi-Touch Gestiken  
Quelle: [Apple Inc. 2011a]

### **Multitasking**

Ab der iOS Version 4 können Apps auch im Hintergrund weiter ausgeführt werden. Diese Funktionalität wird durch Multitasking ermöglicht. Befindet sich eine App im Vordergrund, kann der Nutzer diese durch Betätigung des Home-Buttons in den Hintergrund ablegen. Befindet sich der Anwender auf dem Home-Display, werden durch eine zweifache Ausführung des Home-Buttons alle im Hintergrund abgelegten Apps angezeigt. Nach Auswahl einer Applikation wird diese in den Vordergrund gesetzt und somit wieder angezeigt. Eine Anwendung muss explizit diese Funktionalität unterstützen und umsetzen, damit die App nach Reaktivierung aus dem Hintergrund beim früheren Stand ohne erneute Initialisierung ausgeführt wird. Des Weiteren ist programmtechnisch festzulegen, welche Funktionen im Hintergrund weiter ausgeführt werden sollen [Apple Inc. 23. März 2011, S. 71-72].

### 2.3. Native und plattformübergreifende Entwicklung einer App

Wie bereits in Kapitel 2.1.2 erwähnt, unterscheidet man grundsätzlich zwischen nativen Applikationen und Web-Apps. Der native Code wird auch als Maschinencode bezeichnet. Weiterhin ist der Code das kompilierte Resultat der systemnahen Sprache<sup>15</sup>, die vom Prozessor direkt ausgeführt werden kann und somit hohe Performance ermöglicht. Dementsprechend gestattet eine auf das bestimmte Endgerät und Hersteller zugeschnittene native Programmierung alle Eigenschaften eines Gerätes optimal auszunutzen. Eine native App wird speziell für ein bestimmtes Smartphone-Betriebssystem entwickelt und kann durch den Nutzer auf das Smartphone zusätzlich installiert werden.

Eine Web-Applikation ist dank der zugrunde liegenden Webstandards "weitestgehend" plattformübergreifend. "Weitestgehend" soll hier den Aspekt verdeutlichen, dass eine Web-App nur so weit funktioniert, wie es die Unterstützung und der Funktionsumfang des Browsers auf dem mobilen Gerät zulässt. Denn die Web-Apps werden in einem Fenster des Browsers angezeigt und aufgerufen. Eine Programmierung kann mit einer Kombination aus Hypertext Markup Language (HTML), JavaScript und Cascading Stylesheets (CSS) erfolgen. Idealerweise bietet ein Browser für mobile Endgeräte zusätzliche Funktionen und Attribute an, beispielsweise werden für Web-Apps im Safari des iPhones folgende Einstellungen durch bestimmte HTML-Tags ermöglicht:

```
1 <link rel="apple-touch-icon" href="icon.png"/>
2 <meta name="apple-mobile-web-app-capable" content="yes"/>
```

**Listing 2.1:** iPhone Home Icon für Lesezeichen und Browser Ausblenden

In Zeile 1 des gezeigten Quellcodes wird ein Icon angegeben, das verwendet wird, wenn der Nutzer die Seite als Lesezeichen auf dem Home-Display speichert. Zeile 2 bewirkt, dass nach Auswahl des Lesezeichens die Web-App mit ausgeblendetem Safari Browser gestartet wird. Die App erweckt nun den Anschein, wie eine native App dargestellt zu sein und zu funktionieren, da nicht mehr offensichtlich ist, dass

---

<sup>15</sup>Alternativ kann eine höhere Programmiersprache auch in einen Zwischencode kompiliert und nachfolgend von einem Interpreter ausgeführt werden. Diese Vorgehensweise wird bei Java umgesetzt.

## 2.3. Native und plattformübergreifende Entwicklung einer App

---

sie im Browser abläuft.

Vor der Entwicklung einer mobilen Applikation ist zu entscheiden, ob eine Web-App oder native App als Endprodukt vorliegen soll. Besteht die Forderung, dass der Vertrieb des Produktes in Vertriebskanälen erfolgt, die von den Herstellern des Betriebssystems bereitgestellt werden, ist ein natives Programm Grundvoraussetzung. Ausschließlich auf diese Weise können Apps im Google Android Market oder App Store von Apple angeboten werden [Spiering und Haiges 14. Juni 2010, S. 239]. Schließlich lässt sich eine native Applikation, wie bereits beschrieben, auf dem Endgerät installieren und unterstützt alle vom Gerät bereitgestellten Funktionen. Eine native App ist nicht allein durch die Programmierung in der Systemsprache zu erhalten. Alternativ kann auf ein Framework zurückgegriffen werden, das eine konventionelle und nicht native Sprache durch eine bestimmte Methodik in nativer Form bereitstellt. Folglich brauchen Programmierer nicht über Kenntnisse von Objective-C<sup>16</sup> oder Java<sup>17</sup> verfügen. Stattdessen kann eine native Anwendung auf der Basis von standardisierten Webtechnologien entwickelt werden, wie HTML, CSS und JavaScript [Spiering und Haiges 14. Juni 2010, S. 239]. Auf diese Weise werden Entwickler unterstützt, die nicht bereits mit der nativen Sprache gearbeitet haben und Erfahrung sammeln konnten. Es entfällt hierdurch eine Einarbeitung in die neue native Sprache. Ermöglicht das Framework gar die Unterstützung für verschiedene Betriebssysteme, ist es eine plattformübergreifende Lösung. Das Potenzial dieser Frameworks ist es, als Schnittstelle zwischen einer standardisierten Programmiersprache und mehreren leistungsstarken nativen Sprachen fungieren zu können. Derartige Technologien werden auch als hybride Frameworks und ein erstelltes Produkt als hybride App bezeichnet. Somit ist eine hybride App eine native Applikation, die ursprünglich nicht in der nativen Sprache programmiert wurde. Einige plattformübergreifende Frameworks, die eine Entwicklung von hybriden Apps ermöglichen, werden in Kapitel 2.3.1 vorgestellt.

### 2.3.1. Hybride Frameworks

Im Folgenden wird die wesentliche Arbeitsweise von drei plattformübergreifenden Frameworks beschrieben, die eine Entwicklung von hybriden Produkten ermögli-

---

<sup>16</sup>die native Sprache von iOS

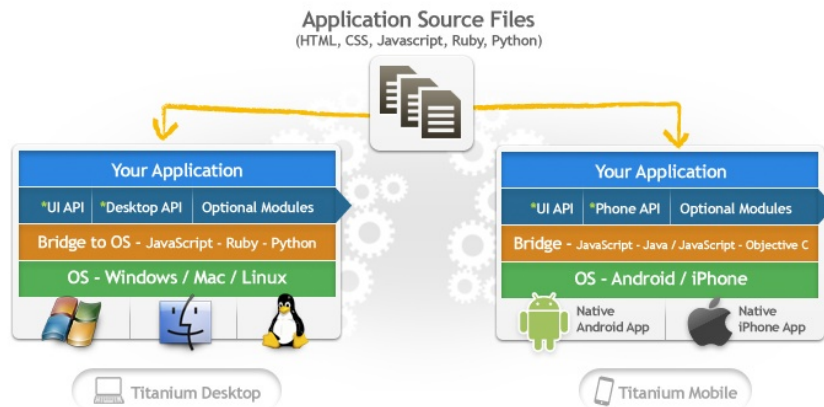
<sup>17</sup>die native Sprache von Android

## 2.3. Native und plattformübergreifende Entwicklung einer App

chen. Ein genauer Vergleich der Funktionen und Möglichkeiten dieser Frameworks wird im Kapitel 3.5 im Verhältnis zu bestehenden Anforderungen durchgeführt.

### Appcelerator Titanium

Titanium Appcelerator wurde von dem Unternehmen Appcelerator Inc. entworfen. Es ist ein Open-Source Framework, mit dem unter Verwendung von Webtechnologien eine native Desktop-, Mobile- oder Tablet-Applikation entwickelt wird (siehe Abbildung 2.7). Titanium wurde im Dezember 2008 [von Fuchs Dezember 2008] veröffentlicht. Seit Juni 2009 [Titanium Juni 2009] wird die Programmierung für



**Abbildung 2.7.:** Plattformübergreifende Entwicklung mit Titanium  
Quelle: [Titanium 2011]

iPhone und Android unterstützt. Laut Appcelerator [Titanium 2011] wird eine Applikation durch das Framework auf folgende Weise zusammengesetzt:

- Die grafische Oberfläche (UI) und die Applikationslogik werden durch HTML, CSS und JavaScript aufbereitet.
- Titanium APIs ermöglichen den Zugriff auf Gerätfunktionen und modulare Elemente mittels JavaScript.
- Eine "language-OS bridge"<sup>18</sup> kompiliert den Web-Code in nativen Code.

<sup>18</sup>Funktionale Einheit, die als Schnittstelle/Brücke fungiert.

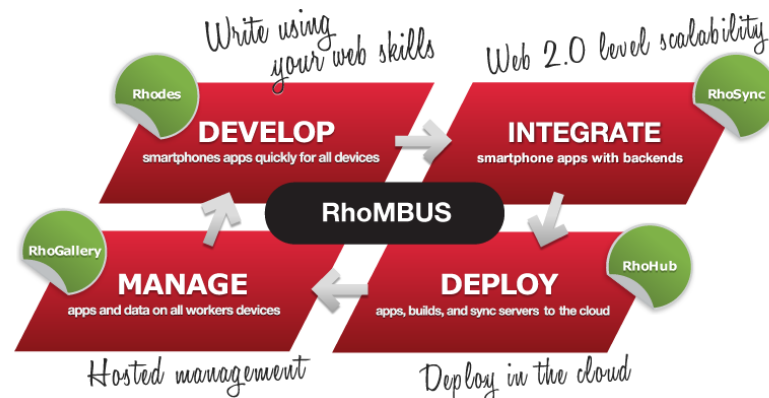
### 2.3. Native und plattformübergreifende Entwicklung einer App

- Die Titanium-Nutzerumgebung<sup>19</sup> formt die Applikation für den Vertrieb<sup>20</sup> und andernfalls für eine Ausführung auf einem Endgerät oder Simulator zusammen.

Von Anfang an ist die mobile App mit der Titanium API zu erstellen, die auf JavaScript basiert. Der Aufbau von Fenstern und Logiken bis hin zum Zugriff auf native Funktionen geschieht in Abhängigkeit der API. Jedes Element wird als native Komponente umgeformt, wodurch die Performance und die Gestaltung der Oberfläche einem nativen Programm entspricht. Ebenso können HTML und CSS als Programmiersprache in Webviews eingesetzt werden.

#### Rhodes

Rhodes ist ein Open-Source Framework von dem Hersteller Rhomobile. Unter der Bezeichnung RhoMBUS vertreibt der Anbieter eine Reihe von Produkten, die den Vertrieb einer mobilen Applikation fördern. Diese Produktreihe wird in Abbildung 2.9 dargestellt.



**Abbildung 2.8.:** Produktsammlung von Rhomobile

Quelle: <http://rhomobile.com/products/> (Zugriff: 25.Juli 2011)

Zusätzlich zu Rhodes wird RhoSync angeboten, das eine serverbasierte Synchronisationsanwendung ist. Für RhoSync muss bei gewerblicher Nutzung eine Lizenz

<sup>19</sup>eng. run-time shell

<sup>20</sup>Es besteht die Möglichkeit eines plattformübergreifenden Vertriebs durch unterschiedliche Pakete z.B. für App Store und Android Market.



### 2.3. Native und plattformübergreifende Entwicklung einer App

---

erworben werden. Wenn nach der Entwicklung einer mobilen App mit Rhodes offline abgelegte Daten in der Anwendung synchronisiert werden sollen, ist RhoSync erforderlich. Rhodes basiert auf der Programmiersprache Ruby<sup>21</sup>, die bei der Anwendung in HTML eingebettet wird. Es werden laut Anbieter die wesentlichen Betriebssysteme von Smartphones unterstützt:

*”Rhodes is an open source Ruby-based framework to rapidly build native apps for all major smartphone operating systems (iPhone, Android, RIM, Windows Mobile and Windows Phone 7).”* [Rhomobile 2011]

#### PhoneGap

PhoneGap ist ein Open-Source Framework und Entwicklungs-Tool, das von Nitobi entwickelt wurde. Es verwendet Webtechnologien wie HTML, JavaScript oder CSS [Nitobi 2011, Vgl. ]. Der wesentliche Unterschied zwischen PhoneGap und den bereits genannten Titanium und Rhomobile ist, dass keine spezifische API verwendet wird, um durch Webtechnologien die Grundbestandteile, wie funktionale Logiken und Aufbau der App, zu erstellen. Stattdessen ist eine Web-App zu entwickeln, die von PhoneGap in einer nativen Webview eingebunden wird. Der Zugriff auf native Funktionen eines Gerätes, wie GPS oder die Kamera, wird durch die PhoneGap API unterstützt. PhoneGap bietet für iOS, Android, Blackberry, WebOS, Windows, Bada und Symbian eine Unterstützung an<sup>22</sup>. Animationen und grafische Darstellungen werden von PhoneGap nicht unterstützt, stattdessen wird auf Entwicklungs-Tools verwiesen<sup>23</sup>, wie das Framework jQuery<sup>24</sup>. Ziel des Anbieters ist es, typische Darstellungen von iOS in ähnlicher Form anzubieten. jQuery basiert auf JavaScript und CSS.

---

<sup>21</sup>Von dem Japaner Yukihiro Matsumoto erstellte Programmiersprache. Bei der Kreation vermischte er Teile seiner Lieblingssprachen (Perl, Smalltalk, Eiffel, Ada und Lisp). Website: <http://www.ruby-lang.org/de/> (Zugriff: 25. Juli 2011)

<sup>22</sup>Komplette Features können auf <http://www.phonegap.com/about/features> (Zugriff 25. Juli 2011) eingesehen werden.

<sup>23</sup>Link zu den von PhoneGap empfohlenen Tools: <http://www.phonegap.com/tools> (Zugriff 25. Juli 2011)

<sup>24</sup>Link zum Framework: <http://jquery.com/> (Zugriff 25. Juli 2011)

## 2.3. Native und plattformübergreifende Entwicklung einer App

### Native versus hybride App

In Abbildung 2.9 werden die Vor- und Nachteile zwischen einer nativen und einer hybriden App vorgestellt.

	Native App	Hybride App
Vorteile	<ul style="list-style-type: none"><li>➤ Hardwarenah</li><li>➤ Aktuelle Bibliothek</li><li>➤ Optimale grafische Darstellung</li><li>➤ Stabil</li><li>➤ App Store Lösung</li><li>➤ UI Standards</li><li>➤ Technischer Support</li><li>➤ Window Builder und viele weitere unterstützende Tools</li></ul>	<ul style="list-style-type: none"><li>➤ Hardwarenah</li><li>➤ App Store Lösung</li><li>➤ Für verschiedene OS mit einer Entwicklung</li><li>➤ Websprache und standardisierte Programmiersprache</li><li>➤ Gewöhnliche UI Elemente für verschiedene OS</li></ul>
Nachteile	<ul style="list-style-type: none"><li>➤ Nur für ein Betriebssystem</li><li>➤ Verschiedene Programmiersprachen müssen beherrscht werden für verschiedene OS</li><li>➤ Nur für einen App Store möglich</li><li>➤ Langsame Entwicklung, unterschiedliche Projekte für verschiedene OS</li></ul>	<ul style="list-style-type: none"><li>➤ Abhängigkeit von Frameworks und ihren Updates</li><li>➤ Limitierter technischer Support</li><li>➤ Limitiertes Debugging</li><li>➤ UI Gestaltung in Abhängigkeit des Frameworks und limitiert</li></ul>

Abbildung 2.9.: Native versus Hybride App

Vor der Auswahl eines hybriden Frameworks besteht die Notwendigkeit, eine umfangreiche Analyse, Überprüfung und Bewertung durchzuführen. Diese Untersuchungen müssen eine Erfüllung von den bestehenden Anforderungen und eine Gewährleistung von den benötigten Funktionen und der Performance validieren. Ergibt sich das Resultat, dass wichtige Voraussetzungen von keinem Framework unterstützt werden, ist eine native Programmierung anzustreben. In diesem Fall gilt es von Anfang an in nativer Sprache zu entwickeln und auf den vollen Funktionsumfang sowie auf die Unterstützung der Betriebssystemhersteller von Smartphones zurückzugreifen. Bei hybriden Apps besteht die Abhängigkeit zu Zweit-Anbietern und den von ihnen durchgeführten Aktualisierungen und erreichten Fortschritten. Der Einsatz von plattformübergreifenden Entwicklungs-Tools wie Rhodes, Titanium oder PhoneGap impliziert eine Abhängigkeit von entsprechenden Anbietern. Wird eine sehr umfangreiche Anwendung programmiert, ist es möglich, dass sich Grenzen des Frameworks

### 2.3. Native und plattformübergreifende Entwicklung einer App

---

aufzeigen. Zum Beispiel gibt es keine Bluetooth Zugriffsmöglichkeiten, native Map Alternative zu Google bei Titanium, Rhodes und PhoneGap oder keine typische UI Unterstützung bei PhoneGap. Des Weiteren muss im Kontext mit einer nativen UI-Unterstützung folgendes beachtet werden: Veröffentlichten Betriebssystemhersteller neue UI-Elemente oder Animationen, können diese von den Plattformen erst nach einiger Zeit durch Updates unterstützt werden. Neue Elemente müssen in den Frameworks übernommen und "nachgebildet" werden. Die Auslagerung von Programmfunktionen in native Webviews ist eine Umsetzung von Funktionen oder Prozessen mit Hilfe von Webtechnologien. Diese Vorgehensweise hat seine Vor- und Nachteile. Wie bereits in Kapitel 2.3 erwähnt, ist die native Sprache die systemnahe Sprache und ermöglicht eine hohe Performance. Die Einbußen der Performance sind somit nachteilig bei der Umsetzung von Funktionen mit Webtechnologien. Ein Vorteil ist, dass eine derartige Auslagerung eine gewisse Unabhängigkeit vom App Store ermöglicht. Neue Updates und Verbesserungen müssen nicht erst von Apple freigeschaltet werden, bevor ein Nutzer diese aktualisieren kann. Stattdessen stehen Verbesserungen dem Anwender umgehend automatisch zur Verfügung, sobald eine externe Webfunktion vom Entwickler angepasst wurde. Ein Update einer Android App wird im Google Market als Download gewertet, wodurch die angezeigte Anzahl von bereits bestehenden Downloads erhöht wird. Einige Nutzer suchen speziell nach Anwendungen, die von vielen Personen verwendet werden (hohe Anzahl von Downloads). Hier zeigt sich ein weiterer Nachteil einer Auslagerung von Funktionen, da durch die automatische Aktualisierung der externen Webfunktionen die Anzahl von Downloads im Android Market nicht erhöht wird.

### 2.4. Verkehrsdatenquellen und Informationsanbieter

In dem Kapitel 2.4.1 wird auf die Erhebung und Verarbeitung von Verkehrsinformationen beim DLR eingegangen und FCD im Kapitel 2.4.2 detailliert erläutert. Zwei alternative Anbieter für mobile Verkehrsinformationsapplikationen in Deutschland werden in dem Kapitel 3.3 genannt.

#### 2.4.1. Verkehrsinformationen des DLR

Am Institut für Verkehrssystemtechnik des DLR e.V. besteht eine Plattform, die als Traffic-Data-Platform (TDP) bezeichnet wird, in der Prozesse und Dienste für Verkehrsdaten verknüpft werden und zusammenwirken. Die Fusion der verschiedenen Komponenten und Informationsquellen ermöglicht eine bessere Interaktion. Die TDP basiert auf dem Systemparadigma einer Serviceorientierten Architektur (SOA). Je nach Anforderungen können autonome Dienste flexibel zur Laufzeit gebunden werden und zusammenwirken [Schmidt 17. Juli 2009, S. 31]. Des Weiteren existieren in der Plattform Komponenten wie Datenverwaltungssysteme und ein Map Server. Es werden Verkehrsdaten, wie Schleifendaten<sup>25</sup>, Floating Car Data (FCD) und Kameradaten<sup>26</sup> in der TDP gesammelt, verarbeitet und analysiert. Die Plattform stellt für Datenbezugsquellen und Dienste spezifische Schnittstellen bereit und ist aus ein oder mehreren Service- und Prozessierungs-Modulen aufgebaut [Schmidt 17. Juli 2009, S. 39-40]. Gerade im Zusammenhang mit FCD ist eine Prozessierung der bezogenen Informationen unabdingbar, da die GPS-Genauigkeit bis zu 13 Metern (siehe Kapitel 2.5.1) abweichen kann. Aufgrund dessen müssen die erhaltenen Positionsinformationen entsprechend aufgewertet und auf ein Straßennetz projiziert werden. Hierfür werden Prozessierungs-Module eingesetzt, die die erhaltenen FCD-Sammlungen entsprechend analysieren und aufbereiten.

Stationäre Detektoren (wie die Induktionsschleife) ermöglichen es, an spezifischen Stellen im Straßennetz die Verkehrsstärke zu bestimmen. Als mobile Detektoren dienen Fahrzeuge durch die FCD erhoben wird. Das System erfordert keine zusätzliche Installation von Sensoren oder von Schleifen auf Fahrwegen, die Installationskosten

---

<sup>25</sup>Induktionsschleifen sind in der Fahrbahn eingelassene Schleifen. Es ist eine Methode für Fahrzeugerkennung und Fahrzeugzählung

<sup>26</sup>Beispielsweise sind auf Brücken und in Tunneln Videokameras angebracht. Durch Bilderkennung und -verarbeitung werden aus den Videodaten verkehrsrelevante Informationen gewonnen.

## 2.4. Verkehrsdatenquellen und Informationsanbieter

---

entfallen. FCD gestattet eine Fahrzeugwiedererkennung an mehreren Messstellen und impliziert dadurch, die Möglichkeit die Reisezeit zu berechnen. Sie bildet eine weitere Messgröße des Verkehrs. Die Technologie zeichnet sich somit durch hohe Bedeutung und Wichtigkeit aus.

### 2.4.2. Floating Car Data (FCD)

Mit FCD (etym. engl.: floating = fließend) sind Informationen gemeint, die im Verkehrsfluss von beteiligten Fahrzeugen erhoben werden. Die Daten ermöglichen eine Berechnung des Verkehrs durch bestimmte Mitwirkende, ohne von jedem Teilnehmer Informationen sammeln zu müssen. Das im Folgenden genannte Basiswissen, wie der Systemaufbau und die Funktionsweise von FCD, ist elementar und exemplarisch für die Umsetzung und Erhebung von FCD in einer mobilen App. Somit gelten die nachfolgenden Erläuterungen als richtungsweisend. Da in dieser Arbeit Verkehrsinformationen für Berlin zur Verfügung gestellt werden, wird speziell auf die Informationsquelle in Berlin eingegangen.

#### Systemaufbau

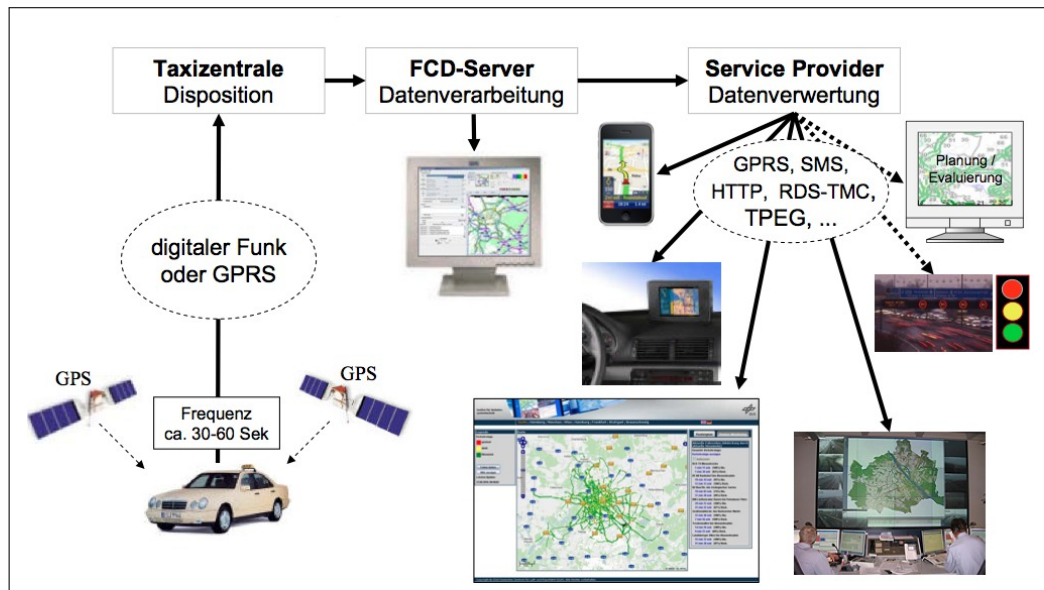
In Abbildung 2.10 wird die Architektur eines FCD-Systems veranschaulicht. Es wird die Struktur des Taxi-FCD-Systems von dem Institut für Verkehrssystemtechnik gezeigt. Ein Systemteilhaber, in diesem Fall ein Taxifahrer, ist mit einem GPS-Empfänger und einem Endgerät ausgestattet, mit dem die empfangenen GPS-Daten über das mobile Internet an eine Zentrale weitergeleitet werden. Die Taxifahrer teilen ihren aktuellen Status der Zentrale mit. Je nach Flottenmanagement-System müssen Fahrer manuell den Status umschalten, oder es wird bei Aktivierung des Taxameters automatisch durchgeführt [Voigt 2009, S. 30-31]. Bei verschiedenen Flottenmanagement-Systemen werden für einen gleichbedeutenden Status unterschiedliche Werte verwendet. In Abbildung 2.11 wird eine Tabelle mit möglichen Zuständen eines Taxis bei den vom DLR separierten Systemen GefoS<sup>27</sup> und Austrosoft<sup>28</sup> gezeigt und die Bedeutung der Zustände erläutert. Diese Statusmitteilungen sind für die FCD-Analyse äußerst wichtig. Es werden nur Positionsdaten für Prognosen von Verkehrssituationen berücksichtigt, bei denen der Taxizustand als besetzt

---

<sup>27</sup>Link zum Anbieter: <http://www.gefos.net> (Zugriff: September 2011)

<sup>28</sup>Internetpräsenz des Unternehmens: <http://austrosoft.com> (Zugriff: September 2011)

## 2.4. Verkehrsdatenquellen und Informationsanbieter



**Abbildung 2.10.:** Systemaufbau des DLR Taxi-FCD  
Quelle: [Brockfeld 24.05.2011]

(“occupied”) vorliegt. In diesem Zustand ähnelt das aktive Fahrverhalten der Taxis den anderen Verkehrsteilnehmern. Eine Abweichung des vergleichbaren Fahrverhaltens liegt durch die potenzielle Nutzung von Busspuren für Taxis vor.

	Austrosoft	Gefos	DLR	Erläuterung
FREE	70	10	20	frei in Umgebung
REGISTERED (LOGGED)	65	11	30	angemeldet
WAITING	87	14,15	40	Pause, wartend beim Kunden
OCCUPIED	66,79,90	30,31,32,34,38	50	besetzt mit Kunde/Fahrziel/Folgeauftrag, besetzt gemeldet
APPROACH (JOURNEY)	75	16	60	"ohne Kunde" Anfahrt zum Kunden, Ziel gemeldet
STAND (HALT)	83	17-20	70	Halteplatz, von Zentrale am Halteplatz eingebucht

**Abbildung 2.11.:** Gesendete Zustände von Taxis und ihre Bedeutung

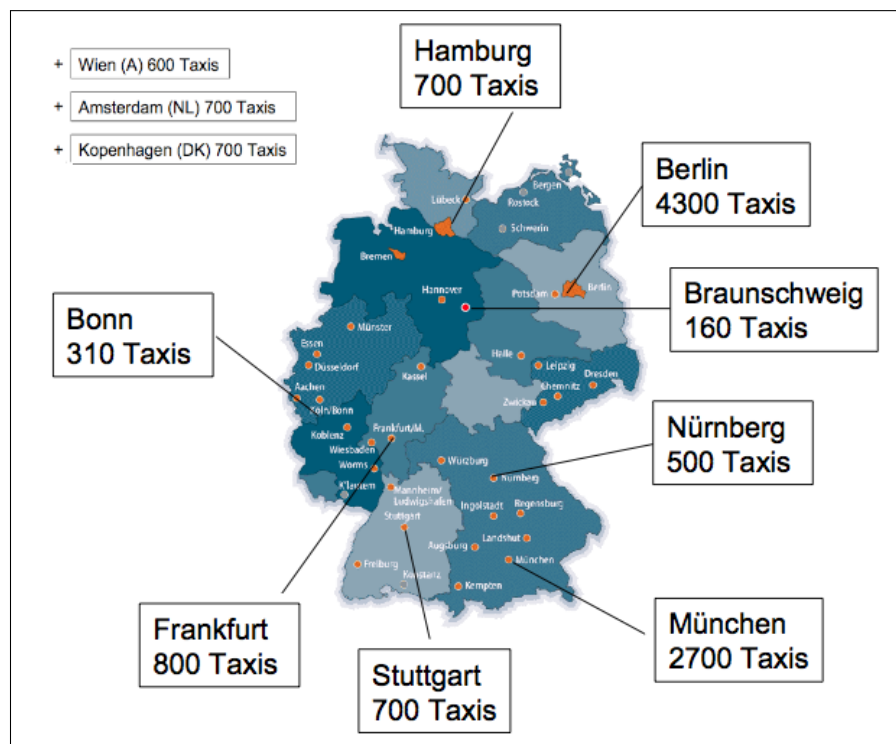
Jedem Taxi wird von der Zentrale eine Fahrzeug-Identifikation (ID) zugeordnet. Diese ID wird in unbekannten Abständen verändert [Voigt 2009, S. 32], um eine Anonymisierung zu ermöglichen und eine Rückverfolgung zu erschweren. Die Taxizentrale übermittelt zusätzlich zu den GPS-Positionsdaten optional die Geschwindigkeit, die ID und den Status des Fahrzeuges an das DLR. In Berlin nehmen mehr als 4300 Taxis an dem System teil, anhand derer die Daten zur Verkehrslage und zu Reisezeiten kontinuierlich ermittelt werden [Sohr u. a. 2010a]. Hierfür besteht

## 2.4. Verkehrsdatenquellen und Informationsanbieter

eine Kooperation mit der Taxizentrale, die die von Taxis erhaltenen Daten an das DLR weiterleitet. Durch den FCD-Server werden die Daten in die TDP eingespeist und bei der Plattform prozessiert und analysiert. Die verschiedenen Schritte und Funktionsweisen der Prozessierung werden in Kapitel 2.4.2 erläutert.

### Abdeckung durch Kooperationspartner (Taxiunternehmen)

Zusätzlich zu den rund 4300 teilnehmenden Taxis in Berlin existieren in Deutschland und Europa weitere Kooperationspartner, die in Abbildung 2.12 mit entsprechender Anzahl von Datenbezugsquellen (Taxifahrer) gezeigt werden. Über 10000 teilnehmende Taxifahrer bilden in Deutschland den Datenbestand des DLR. Dadurch kann das Unternehmen auf eine Rohdatenbasis von mehreren Milliarden Positionsdaten zurückgreifen, die seit 2001 gesammelt wurden [Brockfeld 24.05.2011, S. 10].



**Abbildung 2.12.:** Anzahl von FCD-Fahrzeugen (Taxis) in Großstädten  
Quelle: [Brockfeld 24.05.2011, S. 7]

### Funktionsweise und Prozessierung

Die Prozessierung von FCD ist ein Verkehrsanalysierungs-Verfahren. Das Verfahren wird eingesetzt, um mittels FCD die Reisezeiten und Geschwindigkeiten von Fahrzeugen zu ermitteln und um die groben<sup>29</sup> GPS-Positionen auf ein digitales Verkehrsnetz abzubilden [Brockfeld u. a. 11–15. Juli 2010, S. 1]. Das digitale Kartenetz ist in viele einzelne Kanten (eng. Edge) eingeteilt, welche oftmals auch als Link (Verbindungsglied) bezeichnet werden. Außerdem werden in diesem digitalen Netz zusätzlich zu den Kanten die Länge sowie Start- und Endknoten als Punkt mit Breiten- und Längengrad angegeben. Die Art des digitalen Netzes und weitere Eigenschaften werden in Kapitel 3.1.1 genannt. Im ersten Schritt der Prozessierung erfolgt eine Zuordnung von GPS-Positionen auf Kanten. Dieser Vorgang wird ebenfalls als Matching bezeichnet. Jeder GPS-Position können in einem "engen" Straßennetz durch einen Zuordnungsradius von 30 Metern<sup>30</sup> mehrere Kanten zugewiesen werden, falls keine Eindeutigkeit vorliegt. Durch die Verwendung und Referenzierung von mehreren GPS-Positionen eines Teilnehmers, identifiziert und rekonstruiert der Matcher eine sinnvolle und eindeutige Anordnung zusammenhängender Kanten. Es ist der nachgebildete Bewegungspfad eines Fahrzeuges, der gleichfalls als Trajektorie bezeichnet wird. Im nächsten Schritt der Prozessierung werden die Verkehrsreisezeiten und -geschwindigkeiten und somit die Verkehrslage berechnet. Laut eines Forschungsberichtes wird die Verkehrslage bei der Prozessierung auf folgende Weise beim DLR gewonnen:

*"The resulting data is the current travel speed and a coverage-value for each edge, where data had been captured. Coverage is the distance the vehicle has travelled on the current link, divided by the length of the link. That way, traffic-condition maps for an entire road-network of urban areas are being generated."* [Sohr und Wagner 2008, S. 1]

Die Reisezeit wird als die Differenz zwischen zwei nacheinander folgenden GPS-Sendungen eines Teilnehmers [Sohr u. a. 2010b, S. 2] berechnet. Die Reise-

---

<sup>29</sup>Die Genauigkeit von GPS liegt zwischen 2 bis 13 Meter (siehe Kapitel 2.5.1), somit können erhaltene Positionen von Straßen abweichen.

<sup>30</sup>Dieser Radius wird beim DLR aufgrund jahrelanger Erfahrung favorisiert. Zusätzlich zu der genannten GPS-Ungeauigkeit kann das für die Prozessierung verwendete digitale Netz von anderen Karten abweichen, von denen die Position erhalten wurde. Zum Beispiel durch alte Kartendaten oder unterschiedliche Abmessungen.



## 2.4. Verkehrsdatenquellen und Informationsanbieter

---

geschwindigkeiten von einem Taxi auf einer Sequenz von Kanten, also auf einer Trajektorie mit dem entsprechend bestimmten Abschnitt des Straßennetzes, wird als das Verhältnis der zutreffenden befahrenen Kantenlänge und der Reisezeit berechnet. Die gefahrene Länge im Verhältnis zur Gesamtlänge der Kante ergibt die prozentuale Gewichtung für eine Geschwindigkeit. Der Wert für jede Kante wird anhand des gewichteten harmonischen Mittels gebildet. Auf diese Weise ist es möglich, die stichprobeartigen quantitativen Merkmale der Reisegeschwindigkeiten von mehreren Teilnehmern einer Kante zuzuordnen. Und diese Zuordnung geschieht, gewichtet nach dem Anteil der gefahrenen Strecke, auf dem Link.

Es sei  $n \in \mathbb{N}$  die Anzahl der Geschwindigkeiten,  $g_i \in \mathbb{R}^+$  die Gewichtung,  $x_i \in \mathbb{R}^+$  die entsprechende Geschwindigkeit an dem Index  $i$  mit  $1 \leq i \leq n$ . Dann wird der gewichtete harmonische Mittelwert definiert durch:

$$\bar{x}_{harm} = \frac{\sum_{i=1}^n g_i}{\sum_{i=1}^n \frac{g_i}{x_i}}$$

Eine GPS-Sendung von Teilnehmern ist in regelmäßigen Intervallen notwendig und beim Taxi-FCD-System geschieht dies etwa in einem Zeitabstand zwischen 30 und 60 Sekunden [Sohr u. a. 2010a, S. 3]. Mit zunehmend größeren zeitlichen Intervallen steigt proportional das Risiko einer fehlerhaften Rekonstruktion der befahrenen Kanten und somit auch die Gefahr einer Missinterpretation der Reisegeschwindigkeit auf bestimmten Links [Sohr u. a. 2010b, S. 2]. Genauere Aussagen über die Qualität und Feststellung der geeigneten Zeitintervalle von FCD werden in dem nachfolgenden Kapitel nahegelegt. Die FCD-Echtzeitanalyse zeichnet sich dadurch aus, dass die aufgewerteten Verkehrsinformationen unmittelbar nach der Prozessierung durch Service Provider verschiedenen Diensten bereitgestellt werden.

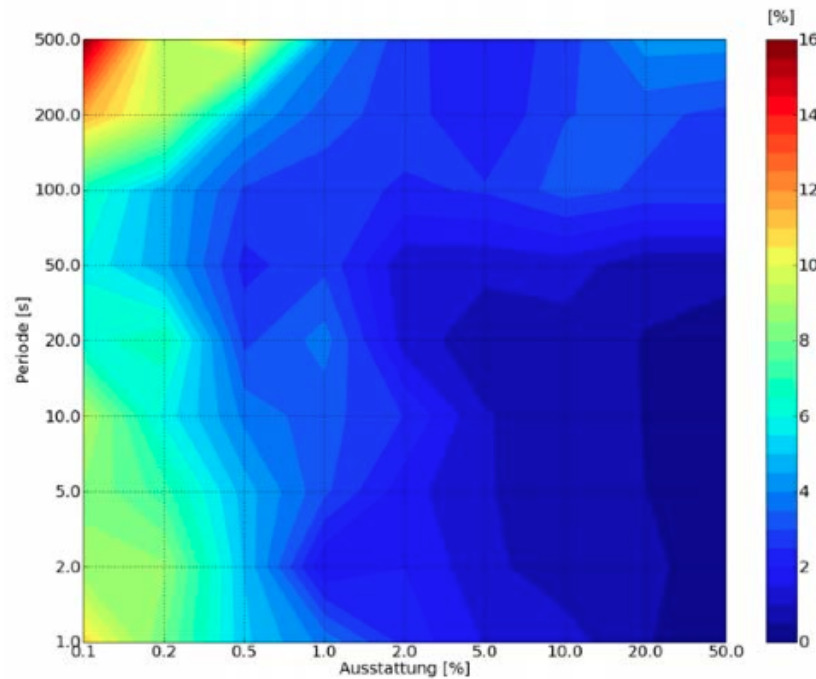
### Qualität

Grundsätzlich ist die Qualität sowie die damit verbundene Aussagefähigkeit und Präzision von FCD als Verkehrsinformation abhängig von der Anzahl der Fahrzeuge und Zeitintervalle zwischen den einzelnen Sendungen [Brockfeld u. a. 11–15. Juli 2010]. Im Zusammenhang mit Fahrzeugen, die an dem FCD-System teilnehmen, spricht man auch von "Equipment Rate", wenn die prozentuale Abdeckung

## 2.4. Verkehrsdatenquellen und Informationsanbieter

---

von Teilnehmern am System im Verhältnis zu allen Verkehrsteilnehmern gemeint ist. In Abbildung 2.13 wird das Ergebnis einer Simulation [Brockfeld u. a. 11–15. Juli 2010] gezeigt. Betrachtet wurde die Abweichung von Geschwindigkeiten zwischen simuliertem Verkehr und daraus erhaltenen FCD-Werten. Hieraus ging her-



**Abbildung 2.13.:** Relativer Fehler im Bezug auf Ausstattungsrate und Intervall  
Quelle: [Brockfeld u. a. 11–15. Juli 2010, S. 5]

vor, dass bereits ab einer Ausstattung von etwa 0.5% und bei Zeitintervallen von Sendungen ab 10 Sekunden gute Ergebnisse erzielt werden können. Es zeigte sich zudem, dass Intervalle unter 10 Sekunden aufgrund redundanter Informationen ungeeignet erscheinen. Des Weiteren wurde bei der Simulation festgestellt, dass eine Prozessierung bei Intervallen über 60 Sekunden zu Ungenauigkeiten führt, da eine Rekonstruktion durch den Matcher nicht mehr möglich ist.

## 2.5. Positionsbestimmung eines Smartphones

Im Folgenden werden die Technologien genannt, mit deren Anwendung eine geographische Information des mobilen Endgerätes erhalten wird. Entsprechend der technischen Beschreibung von Apple [Apple Inc. 2011b] werden für eine Ortung des iPhones folgende Methoden eingesetzt:

- Assisted Global Positioning System (A-GPS)
- Digitaler Kompass
- Wi-Fi
- Mobilfunknetz

Laut Apple [Apple Inc. Januar 2011] arbeitet der digitale Kompass ähnlich wie ein üblicher Magnetnadel-Kompass. Der digitale Kompass dient bei einer Positionsbestimmung lediglich als Präzisierung von der Lokalisierung durch A-GPS, Wi-Fi und Mobilfunk. Aufgrund dessen wird in den folgenden Kapiteln auf diese drei Ortungsmethoden eingegangen. Für eine Mobilfunkortung ist die Positionskennung von Sendemasten und für eine Wi-Fi-Ortung sind die geografischen Informationen von Accesspoints elementare Voraussetzung. Eine Identifikation der Sendeeinheit sowie deren geografische Lage (zum Beispiel Breiten- und Längengrad (eng. Latitude and Longitude)) und optionale ortsgebundene Zusatzinformationen (beispielsweise Straße, Hausnummer, Postleitzahl) müssen in einer Datenbank vorhanden sein, um mit diesen Methoden eine Lokalisierung durchführen zu können. In dem Kapitel 2.5.4 wird diesbezüglich die Vorgehensweise von Apple aufgezeigt.

### 2.5.1. Positionsermittlung durch das Global Positioning System (GPS)

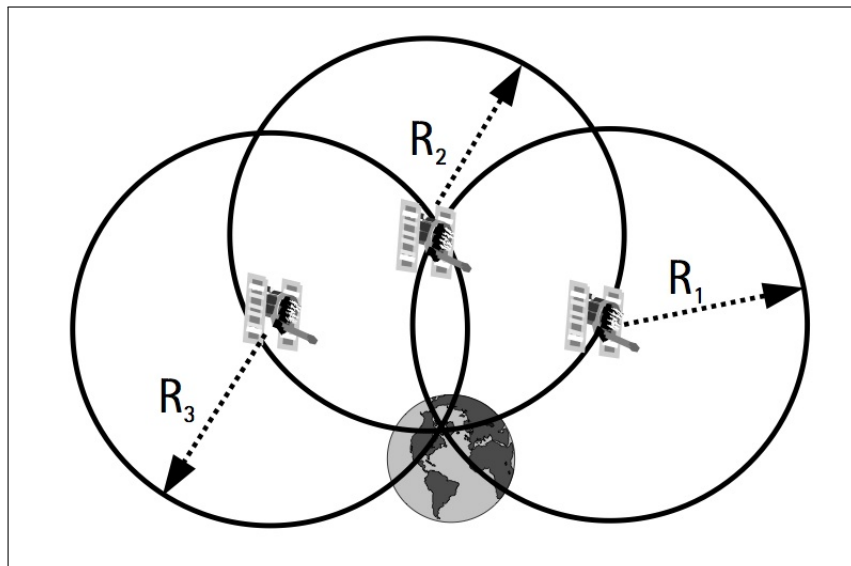
Im Folgenden wird die Positionsbestimmung durch GPS beschrieben, die für diese Ausarbeitung wesentlich ist. Für den interessierten Leser werden weitere Informationen über GPS zusätzlich im Anhang B bereitgestellt.

Für die Lokalisierung des GPS-Nutzers ist es essenziell, dass die genaue Position von empfangenen Satelliten bekannt ist. Nachfolgend wird die minimale Anzahl von benötigten und "sichtbaren" Satelliten genannt und erläutert. Zum besseren Verständnis wird zuerst die Positionsermittlung vereinfacht und abstrakt erklärt:

## 2.5. Positionsbestimmung eines Smartphones

Es ist davon auszugehen, dass die Position von empfangenen Satelliten vorliegt und bekannt ist. Die Entfernung des Empfängers zu einem Satelliten ergibt sich aus der Signallaufzeit. Die Entfernung kann als Radius einer Signalkugel angesehen werden, die von Satelliten ausgestrahlt wird. Erhält man das Signal zu drei Satelliten und bildet zu diesen die Kugeln, schneiden sie sich in einem Punkt. Dieser Punkt bildet die Position des Empfängers.

Nachfolgend wird in diesem Kapitel die vereinfachte Erklärung präzisiert. In Abbildung 2.14 ist die grundsätzliche Funktionsweise der Positionsermittlung eines GPS-Empfängers abgebildet. Das System führt die Positionsbestimmung mithilfe des



**Abbildung 2.14.:** Die Basisidee der Positionsbestimmung beim GPS  
Quelle: [El-Rabbany 2002, S. 9]

Prinzips von Time of Arrival (TOA) bei Funksignalen aus. Die benötigte Übertragungszeit multipliziert mit der Geschwindigkeit des Signals, ergibt die zurückgelegte Sendestrecke. In dem Vakuum des Weltalls breitet sich das Signal des Satelliten mit Lichtgeschwindigkeit ( $299.792.458 \frac{m}{s}$ ) aus. In Anhang B.1 wird genannt, dass sehr genaue<sup>31</sup> Atomuhren in den Satelliten eingesetzt werden. In der Praxis können in

<sup>31</sup>mindestens auf die  $10^{-13}$  Sekunde genau [Köhne und Wößner 2007] [El-Rabbany 2002, S. 31]

## 2.5. Positionsbestimmung eines Smartphones

---

den GPS-Empfängern keine hochgenauen Atomuhren eingebaut werden<sup>32</sup>, stattdessen werden Quarzuhren verwendet<sup>33</sup>. Bei fehlerhafter Messung der Zeit kann dies zu groben Ungenauigkeiten der Positionsmessung führen. Im Folgenden wird ein Rechenbeispiel, das schon bei einer geringfügig fehlerhaften Zeitmessung von einer Mikrosekunde, die potenzielle Fehlerhaftigkeit der Positionsmessung verdeutlicht:

$$\mathbf{u} = a \cdot c$$

$$\mathbf{u} \approx 299m$$

Abweichung der Zeit  $a = 0.000001s$  (In Sekunden)

Lichtgeschwindigkeit  $c = 299.792.458 \frac{m}{s}$  (In Meter pro Sekunde)

Ungenauigkeit der Messung  $u$  (In Meter)

Der Empfänger kalkuliert die Signallaufzeit, indem die eigene Uhrzeit von dem mitgesendeten Zeitstempel des Satelliten subtrahiert wird. Anhand des bereits beschriebenen TOA wird die zurückgelegte Signalstrecke wie folgt berechnet:

$$Range_S = (t_E - t_S) \cdot c$$

$Range_S$  : Entfernung von Satellit S zum Empfänger

$t_E$  : Zeit des Empfängers bei Signalankunft

$t_S$  : Zeit des Satelliten bei Signalerzeugung / Signalsendung

Solange die genannte Uhrenungenauigkeit des Empfängers mit der genauen Zeit der Satelliten nicht synchronisiert ist, werden die gemessenen Entfernungen auch als Pseudorange<sup>34</sup> bezeichnet [El-Rabbany 2002, S. 19-20]. Für eine 3D-Positionsbestimmung<sup>35</sup> im Raum muss die Pseudoentfernung von mindestens vier empfangenen Satelliten berechnet werden. Denn es liegen zu den drei unbekannten Koordinaten der Empfängerposition  $x, y, z$  zusätzlich  $\Delta t$ , die unbekannte Abweichung der Empfängeruhr zur Satellitenzeit, vor. Es wird für jeden Satelliten folgende Gleichung aufgestellt:

$$Range_S = \sqrt{(x_E - x_S)^2 + (y_E - y_S)^2 + (z_E - z_S)^2} + \Delta t \cdot c$$

---

<sup>32</sup>Es werden Kosten als Grund genannt [Köhne und Wölkner 2007].

<sup>33</sup>etwa  $10^{-3}$  Sekunden genau

<sup>34</sup>Entfernung im Englischen Range

<sup>35</sup>Bestimmung von Längen- und Breitengrad sowie Höhe

## 2.5. Positionsbestimmung eines Smartphones

---

Bei vier Satelliten ergibt sich ein nichtlineares Gleichungssystem mit vier Gleichungen und vier Unbekannten. Die Lösungsmenge kann vom GPS-Empfänger numerisch durch Approximation und Iteration kalkuliert werden. Die Position  $(x, y, z)$  und die Uhrenabweichung des Empfängers sind berechnet, somit ist die Empfängeruhr mit der genauen Satellitenzeit synchronisiert. Bei nachfolgenden Positionsbestimmungen, beispielsweise bei einer Navigation, reicht der Empfang von drei Satelliten aus.

### GPS Genauigkeit und Techniken zur Verbesserung/Unterstützung

Herkömmliche GPS-Empfänger erzielen eine Genauigkeit von 2-13 Metern [Elektronik Kompendium]. Eine Genauigkeit von unter zwei Metern ist mit Differential-GPS erreichbar. Hierbei werden geografisch nahe Referenzstationen genutzt, die z.B. Ionosphäreinformationen über Funk<sup>36</sup> für eine Korrektur bereitstellen [Zogg 2002, S. 48-51].

A-GPS (Assisted Global Positioning System) ist eine Verfahrenstechnik, durch die eine schnellere und genauere GPS-Positionierung ermöglicht werden kann. Dabei werden unterstützende Informationen, wie zum Beispiel die Almanach Daten oder Lokalisationsinformationen durch Mobilfunkortung, über ein Datennetz gesendet. A-GPS findet überwiegend bei modernen Mobiltelefonen (Smartphones) seine Anwendung [Karunanayake u. a. 2004]. Durch die Übertragung der Informationen können jedoch Kosten für den Nutzer entstehen<sup>37</sup>.

### 2.5.2. Mobilfunkortung

Anfang des Jahres 1980 entstand bei der Franco-German Kooperation die Initiative für ein digitales Kommunikationssystem, das darauf folgend als Group Spéciale Mobile in der Conférence Européenne des Administrations des Postes et des Télécommunications (CEPT) eingeführt, sowie später mit der Gründung des European Telecommunication Standards Institute (ETSI) als Global System for Mobile Communication (GSM) kreiert wurde [Eylert 2005, S. 12]. GSM ist ein Nachfolger der analogen Systeme der ersten Generation und ist der Mobilfunkstandard der 2. Generation (2G) [Eylert 2005, S. 2]. Das digitale Kommunikationssystem wurde im Jahr 1992 eingeführt [Gutmann Juni 2007, S. 3] und ist seither der weltweit verbreitetste

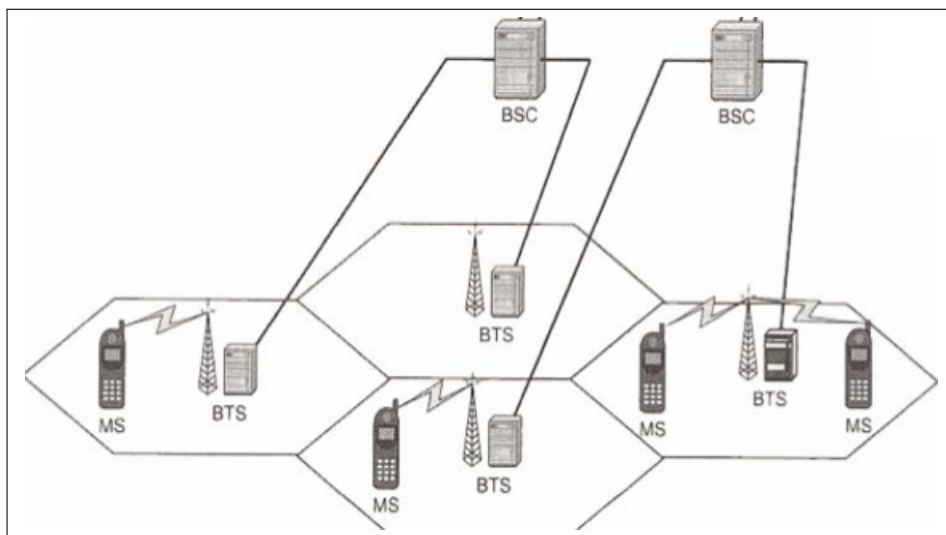
---

<sup>36</sup>Lang-, Mittel- und Kurzwellenbereich

<sup>37</sup>In Abhängigkeit von dem Tarif und Mobilfunkanbieter

## 2.5. Positionsbestimmung eines Smartphones

Mobilfunkstandard. Jedem Mobilfunkteilnehmer ist global eine eindeutige Identifikation durch die International Mobile Subscriber Identity (IMSI) zugeordnet, die nach ITU-T E.212 Standard definiert ist [Eylert 2005, S. 136] und auf der SIM (Subscriber Identity Module) Karte vor Ausgabe an den Nutzer gespeichert wird. Diese eindeutig weltweite Identifikation wird von Mobilfunk Providern vergeben und beinhaltet den Mobile Country Code (MCC)<sup>38</sup>, Mobile Network Code (MNC)<sup>39</sup> und Mobile Subscriber Identification Number (MSIN)<sup>40</sup> mit maximal 15 Ziffern [Eylert 2005, S. 136-137]. Das Mobilfunknetz besteht aus Funkzellen. Eine bildliche Abstraktion des Systemaufbaus wird in Abbildung 2.15 gezeigt. Es handelt sich um



**Abbildung 2.15.:** Abstraktion des Systemaufbaus vom Mobilfunknetz  
Quelle: [Friedrich 31. Oktober 2007]

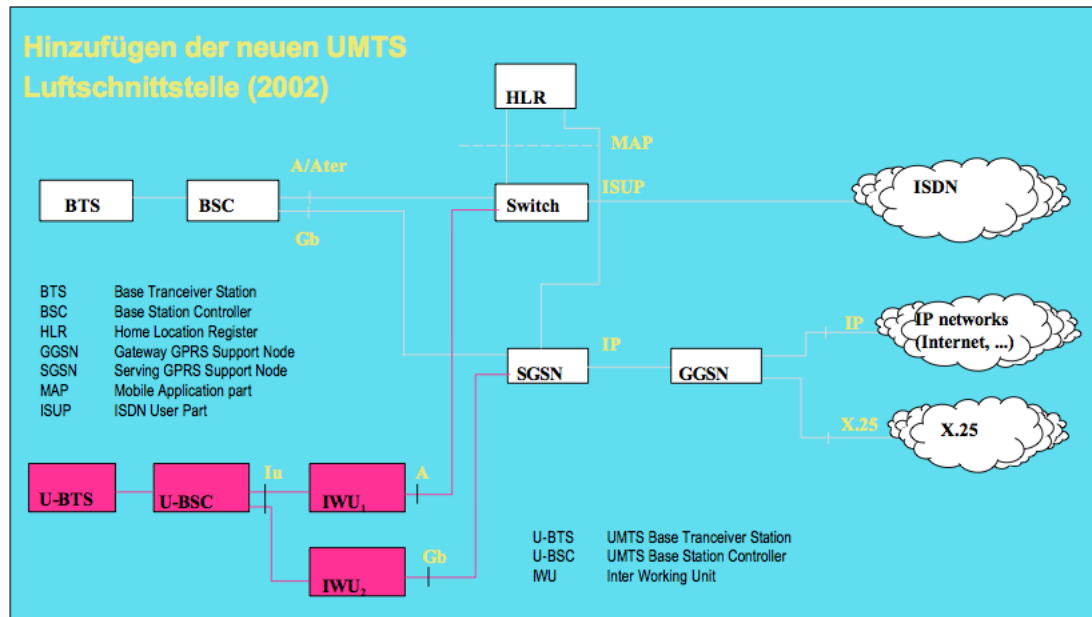
eine Abstraktion, da die Grafik zum einen den Systemaufbau nur grob zeigt und zum anderen die Zellenstruktur in Realität nicht derartig einheitlich ist. In Wirklichkeit ist die Ausdehnung der Zellen von einigen Randbedingungen, wie durch die Netzdichte und die Abschirmung von Gebäuden, abhängig. Eine weitere Abbildung 2.16 veranschaulicht zwar nicht die Zellenabstraktion, dafür werden hier aber weitere Komponenten des GSM gezeigt, die für eine Mobilfunkortung relevant sind.

<sup>38</sup>Besteht aus 3 Ziffern, zum Beispiel für Deutschland 262.

<sup>39</sup>MNC ist maximal 3 Ziffern groß.

<sup>40</sup>MSIN kann aus mehreren Ziffern bestehen, jedoch höchstens aus 10 Ziffern. Darf nur dann aus 10 Ziffern bestehen, wenn die MNC mit weniger 3 Ziffern gewählt wird.

## 2.5. Positionsbestimmung eines Smartphones



**Abbildung 2.16.:** GSM Komponenten mit UMTS Luftschnittstellen  
Quelle: [Eylert WS 2007]

Ebenso werden in dieser Grafik die im Jahr 2002 hinzugefügten Luftschnittstellen für Universal Mobile Telecommunication System (UMTS)<sup>41</sup> dargestellt. Weiterhin zeigt die Abbildung den Übergang von GSM zu General Packet Radio Service (GPRS)<sup>42</sup> durch Gateway GPRS Support Node (GGSN) und Serving GPRS Support Node (SGSN). Bei GSM wird jede Funkzelle von einer Base Transceiver Station (BTS) versorgt. Eine Überwachung und Steuerung von BTS erfolgt über die Base Station Controller (BSC). Dem BSC wird vom Provider eine eindeutige Location Area Identity (LAI) zugewiesen. Sie besteht aus Mobile Country Code (MCC), Mobile Network Code (MNC) und Location Area Code (LAC). Des Weiteren ordnen die Provider den Basisstationen eine in ihrem Netzwerk eindeutige Zellenidentifikationskennung zu. Das Mobile-services Switching Centre (MSC) verknüpft mehrere BSC miteinander und übernimmt wesentliche Aufgaben, wie ein- und ausgehende Anrufe entsprechend weiterzuleiten oder mit dem Festnetz eine Verbindung herzustellen

<sup>41</sup>Es ist der Mobilfunkstandard der 3. Generation (3G) für Multimedia-Dienste und schnellere Übertragungsraten [Gutmann Juni 2007, S. 2].

<sup>42</sup>Paketorientierter Dienst für die Datenübertragung, der in GSM einen Zugang zum Internet ermöglicht [Gutmann Juni 2007, S. 6].



## 2.5. Positionsbestimmung eines Smartphones

---

[Mattsson 2001, S. 7-9]. Wählt sich ein Mobiltelefon in das Netz einer BTS ein, überträgt es die IMSI. Die IMSI wird im Home Location Register (HLR) gespeichert und auf einem Eintrag im Visitor Location Register (VLR) referenziert. Das Home Register ist ein globales Register von Betreibern. Es enthält den aktuellen Aufenthalt und die Location Area aller Teilnehmer des Providers [Mattsson 2001, S. 8]. Hingegen ist das VLR ein regionales und einer MSC zugeordnetes Register, indem die festgestellte Location Area und Mobilfunknummer des Teilnehmers zusätzlich zu der IMSI gespeichert sind [Gutmann Juni 2007, S. 5]. Eine Aktualisierung findet in dem Register statt, wenn der Mobilfunkteilnehmer einer anderen Region zugeordnet wird. Die Ortung des Mobilfunkgerätes kann durch eine Bestimmung der Zellenidentifikation<sup>43</sup> geschehen, wobei eine Präzisierung beispielsweise durch die Hinzunahme und Berechnung von Signalstärken und -laufzeiten erreicht wird. Die Zellenidentifikation ist zusammen mit der LAI global eindeutig und diese werden einem mobilen Endgerät bei bestehender Verbindung von einer BTS übertragen. Im Kollektiv ermöglichen die Kennungen eine Bestimmung des Aufenthaltes, wenn zu diesen eine Referenzierung auf geografische Positionsinformationen vorliegt. Die Genauigkeit der Ortung hängt von der Größe einer Zelle ab und kann in ländlichen Gebieten sogar bis zu 35 Kilometer [Mattsson 2001, S. 20] und in der Stadt etwa 100 Meter betragen [Gutmann Juni 2007, S. 4].

### 2.5.3. Wi-Fi

Wireless Fidelity (WiFi/Wi-Fi) gilt als Synonym für WLAN in Amerika [Eylert 2005, S. 70]. Ein WLAN-Zugangspunkt verfügt über einen international eindeutigen Basic Service Set Identifier (BSSID), der nach dem Standard IEEE 802.11-1999 [IEEE-Standards 2010] sowie nach Revision IEEE 802.11-2007 [IEEE802.11 2007] als die Medium Access Control (MAC), eine 48 Bit große Adresse, definiert ist. Die Stationen senden unter anderem ihre MAC-Adresse und den nicht eindeutigen und frei wählbaren Service Set Identifier (SSID)<sup>44</sup> aus<sup>45</sup> [IEEE802.11 2007, S. 318]. Deshalb ist die MAC-Adresse von empfangenen WLAN-Stationen für Endgeräte ermittelbar. Wenn geografische Positionsdaten anhand einer MAC zur Verfügung

---

<sup>43</sup>Wird auch als Cell of Origin (COO) bezeichnet.

<sup>44</sup>Es ist der Name des Funknetzes.

<sup>45</sup>Hier besteht die Annahme, dass die SSID Sendung aktiviert ist. Einige Zugangspunkte verfügen über die Funktionalität, eine SSID Sendung zu deaktivieren.

## 2.5. Positionsbestimmung eines Smartphones

---

stehen<sup>46</sup>, kann eine Positionsermittlung durchgeführt werden. Nach IEEE 802.11 Norm wurde WLAN 1997 zum ersten Mal spezifiziert [Winkler 2009]. Zu diesem ursprünglichen Standard liegen bis heute einige Erweiterungen vor. Beim iPhone 4 [Apple Inc. 2011b] werden 802.11b/g/n Wi-Fi unterstützt. Die Reichweite von WLAN ist stark von Hindernissen (zum Beispiel Gebäudewände) abhängig. Bei der Indoor-Verwendung kann die Reichweite bis zu 50 Metern und im Freien bis zu 300 Metern betragen [Teker November 2005, S. 29].

### 2.5.4. Details zur Ortung bei Apple Produkten

Wie bereits erwähnt, ist für eine Mobilfunk- oder Wi-Fi-Ortung die Kennung der Position von Sendemasten oder WLAN-Geräten die Grundvoraussetzung. Diese müssen auf geeignete Weise erhoben und gespeichert werden. Location-Based-Services werden laut Apple den Kunden folgendermaßen angeboten:

*”To provide location-based services on Apple products, Apple and our partners and licensees may collect, use, and share precise location data, including the real-time geographic location of your Apple computer or device.”* Apple Inc., [Apple Inc. Juni 2010]

Demzufolge gibt Apple bekannt, dass sie, die Partner und Lizenzbesitzer (Developer), Positionsdaten sammeln, verwenden und gemeinsam teilen. Ab der Betriebssystemversion 4 des iOS werden bei 3G-fähigen Geräten kontinuierlich Ortungsinformationen von Wi-Fi und Mobilfunk in der Datei ”consolidated.db” gesammelt, bei iOS Vorgängerversionen geschieht dies in der Datei ”h-cells.plist” [Levinson April 2011] [Levinson u. a. 2011]. Die ”h-cells.plist” Datei wird verborgen und ist für andere Applikationen nicht zugänglich. Hingegen wird die ”consolidated.db” Datei bei einer Synchronisation mit dem PC als Backup auf den Computer übertragen [Levinson April 2011] und ist für Location-Based-Services Anbieter unter Vorbehalt zugänglich. Dieser Sachverhalt sorgte im April 2011 in den Medien für große Unruhe und wurde unter anderem mit Schlagzeilen wie ”Apple überwacht Besitzer”<sup>47</sup> betitelt. Apple reagierte mit einer ausführlichen Stellungnahme:

---

<sup>46</sup>Beispielsweise kann die Sammlung und Zuordnung von MAC-Adressen auf Positionsinformationen in einer Datenbank erfolgen.

<sup>47</sup>Link zum Artikel der Nachrichtenagentur Tageblatt: <http://www.tageblatt.lu/nachrichten/story/26124748> (Zugriff: Juli 2011)

## 2.5. Positionsbestimmung eines Smartphones

---

*”The iPhone is not logging your location. [...] a crowd-sourced database of Wi-Fi hotspot and cell tower data that is generated by tens of millions of iPhones sending the geo-tagged locations of nearby Wi-Fi hotspots and cell towers in an anonymous and encrypted form to Apple.”* Apple Inc., [Apple Inc. 27. April 2011]

Das Unternehmen betont die Notwendigkeit dieser Datenerhebung für die Funktionalität von Location-Based-Services bei Apple Produkten und versichert zudem eine anonymisierte und verschlüsselte Übertragung an ihre Server. Durch die Einführung von Multitasking (siehe Kapitel 2.2.2) war das Offenlegen von Ortungsinformationen in der *”consolidated.db”* auf diese Weise für Apple unabdingbar, um Applikationen von Dritt-Anbietern ebenso bei einer Ausführung im Hintergrund, die Sammlung von Ortungsdaten bei Bedarf bereitstellen zu können. Ab der iOS Version 4 sind die Ortungsinformationen für Programme nur unter Vorbehalt zugänglich, der Nutzer muss explizit die Verwendung der Daten für ein Programm erlauben. Wird eine Anwendung zum ersten Mal gestartet, erscheint für den Nutzer eine Anfrage. Exemplarisch wird eine Ortungsanfrage durch die Karten-App von Google in Abbildung 2.17 dargestellt. Die Zugriffsoptionen auf Ortungsdaten von einzelnen Programmen kann der Benutzer auch in den Einstellungen des iOS modifizieren (*Einstellungen > Allgemein > Ortungsdienste*) [Apple Inc. 2010b]. Verweigert der Anwender den



**Abbildung 2.17.:** iOS Abfrage der Ortungserlaubnis  
Quelle: [Apple Inc. 2010b]

Zugriff auf die Lokalisierungswerte, kann ein Programm die Informationen nicht verwenden. Eine weitere Ansammlung von Ortungsinformationen in der Logdatei

## 2.5. Positionsbestimmung eines Smartphones

---

”consolidated.db” wird trotz Deaktivierung von Lokalisierungsdiensten durchgeführt [Levinson April 2011].

### Technische Funktionsweise

Für den Zugriff und die Steuerung der Ortungsinformationen wird von Apple ein Core Location Framework [Apple Inc. 2010c, S. 12] bereitgestellt. Mit der Klasse ”CLLocationManager” bietet Apple eine Schnittstelle an [Apple Inc. 2010a], die den Aufenthalt berechnet, Lokalisierungsinformationen zur Verfügung stellt und die Zugriffsberechtigung einer Applikation auf die Positionsdaten steuert.

### 2.5.5. Alternative Ortungsmöglichkeiten

Die Ortung erfolgt bei verbreiteten Smartphones<sup>48</sup> - wie bereits in Kapitel 2.5 erläutert - mit den Methoden GPS, WLAN und Mobilfunk. Alternative Globale Navigationssatellitensysteme (GNSS) zu GPS sind beispielsweise das russische Satellitensystem GLONASS<sup>49</sup> und das europäische Satellitensystem Galileo<sup>50</sup>. Galileo befindet sich im Entwicklungsstadium und ist voraussichtlich ab 2014 als Satellitenortungssystem einsatzfähig [European Commission 2011]. In dem Kapitel 2.5.4 wurde nahegelegt, wie bei iOS eine Ortsbestimmung durchgeführt wird. Die mobile Anwendung ist unabhängig davon, welche Methoden verwendet werden. Je nachdem, welche Ortungsmethoden das Gerät und das Location Framework unterstützen, wird die Position ermittelt und die berechneten Daten den mobilen Applikationen zur Verfügung gestellt. Die Anwendung kann somit ein anderes GNSS oder eine alternative Ortungsmethode<sup>51</sup> für die Positionsermittlung verwenden, wenn das Gerät und die native Funktion der Lokalisierung diese ermöglichen.

---

<sup>48</sup>in dieser Arbeit am Beispiel des iPhones

<sup>49</sup>Es ist ein GNSS, das von dem Verteidigungsministerium der Russischen Föderation betrieben wird. Folgende Website des Information-Analytical centre of coordinate-time and navigation support (IAC CTNS) zeigt für Privatanutzer unter anderem GLONASS Status und Almanach Daten auf: <http://www.glonass-ianc.rsa.ru/en/> (Zugriff: September 2011)

<sup>50</sup>Galileo ist eine Initiative der European Commission (EC) - Website: [http://ec.europa.eu/enterprise/policies/satnav/galileo/index\\_en.htm](http://ec.europa.eu/enterprise/policies/satnav/galileo/index_en.htm) und der European Space Agency (ESA) - Internetpräsenz: <http://www.esa.int/esaNA/galileo.html> (Zugriff: September 2011)

<sup>51</sup>Zum Beispiel eine Ortung mit Bluetooth, siehe Website: <http://indoor-ortung.de/technik/ortung-mit-bluetooth> (Zugriff: September 2011)

---

## 3. Systematische Analysen

### 3.1. Ist- und Sollzustand

#### 3.1.1. Istzustand

##### **Routing-Engine und Routen-Monitoring**

Für die Überwachung von Routen<sup>1</sup> wird auf die Funktionalität eines DLR Dienstes zurückgegriffen, der intern als "Travel-Time-Service" bezeichnet wird. Dieser Service berechnet für eine Route Informationen über die aktuelle Verkehrsreisezeit und Streckenlänge. Des Weiteren bestimmt der Dienst für die Route eine bestehende historische Reisezeit. Die historischen Werte werden folgendermaßen kalkuliert: Für jede Stunde eines Tages wird die durchschnittliche gefahrene Geschwindigkeit auf einer Kante des digitalen Kartennetzes durch die Verwendung von Verkehrsdaten der letzten drei Wochen berechnet. Ebenfalls gibt der Service über auftretende Zeitverzögerungen sowie über die Streckenlänge der Verzögerung und deren Anfangs- und Endposition Auskunft. Diese Verzögerungen werden berechnet, indem die aktuelle Verkehrslage mit einer "angenommenen" verkehrsgünstigen Lage im Zeitraum von 0-5 Uhr nachts auf der Strecke verglichen wird. Zu dieser Annahme wird in Kapitel 7.4 eine persönliche Einschätzung beschrieben. Die Kalkulation von verkehrsoptimierten Routen ermöglicht der DLR Dienst mit einer Routing-Engine<sup>2</sup> für Berlin (und teilweise Umland). Dieser Service umfasst folgende Funktionalitäten: Berechne Straße und Postleitzahl für eine gegebene Position, liefere Position für eine gegebene Straße und Postleitzahl, liefere alle möglichen Straßen für eine gewisse Zeichenkombination (zum Beispiel "Magn"), berechne für eine gegebene Start- und Zielposition verkehrsoptimiert eine Route. Die Funktionalitäten der Routing-Engine

---

<sup>1</sup>Die Methode "Überwachung von Routen" wird auch als "Routen-Monitoring" bezeichnet.

<sup>2</sup>Es ist eine Funktionseinheit zum Berechnen von Routen.

### 3.1. Ist- und Sollzustand

---

und des Routen-Monitorings bestehen zwar, fanden sich aber parallel zu dieser Masterarbeit in einer Weiterentwicklung und Modernisierung. Bisher wurde der Zugriff auf diese Funktionalitäten intern über Sockets<sup>3</sup> vollzogen und auf diese Weise eine Verbindung zu entsprechenden Diensten aufgebaut. Mit der Modernisierung wird Apache ActiveMQ<sup>4</sup> beim DLR eingesetzt, um intern in verschiedenen Projekten auf die Funktionalität der Services und Applikationen zugreifen zu können. Apache ActiveMQ ist eine Open-Source Implementation von Java Message Service(JMS)<sup>5</sup> und ermöglicht das Senden und Empfangen von Nachrichten in verteilten Anwendungen. Das MQ (Message-Queuing) funktioniert dabei folgendermaßen: Die Dienste melden sich mit ihrer eindeutigen Identifikation (beim DLR eine Zeichenkette als Name) bei dem Sendesystem (Message Broker) an. Der Message Broker von Apache ActiveMQ verwaltet die Dienste in einer Queue. Möchte ein Client an einem Service eine Anfrage stellen, muss er zuerst eine Verbindung mit Angabe der Identifikation des Dienstes zum Broker aufbauen. Sendet der Client nachfolgend eine Nachricht an den Service - beispielsweise eine Anfrage für eine Route - leitet der Message Broker die Nachricht an den Service weiter. Zudem wird ebenfalls eine eindeutige Kennung für den Client erzeugt. Der Service erstellt auf diese Anfrage eine Antwort und gibt zusätzlich die Kennung für den Client an. Die Antwort ist nur für den Client vorgesehen und wird von dem Broker gesendet.

Die genannte parallele Modernisierung stellt diese Masterarbeit vor eine zusätzliche Herausforderung, da einige benötigte Funktionalitäten zum Zeitpunkt der Entwicklung nicht umgesetzt waren. Trotzdem wird im Rahmen dieser Masterarbeit auf die Modernisierung zurückgegriffen, um nach vollständiger Weiterentwicklung und Erneuerung der Dienste in Zukunft, die neue Funktionalität zu unterstützen.

#### Digitale Straßenkarten

Wie bereits in Kapitel 2.4.2 erwähnt wurde, bilden digitale Straßenkarten die Grundlage zur Prozessierung von FCD und zur Analyse der Verkehrslage. Das Straßennetz ist in viele Kanten separiert. Beim DLR wird hierfür das Kartenmaterial des Unter-

---

<sup>3</sup>Socket ist eine plattformunabhängige Software-Schnittstelle für die Kommunikation und den Datenaustausch im Netzwerk.

<sup>4</sup>Internetpräsenz des Anbieters: <http://activemq.apache.org/> (Zugriff: August 2011)

<sup>5</sup>Link zu einem technischen Artikel über JMS: <http://java.sun.com/developer/technicalArticles/Ecommerce/jms/> (Zugriff: August 2011)

### 3.1. Ist- und Sollzustand

nehmens NAVTEQ<sup>6</sup> verwendet. Jede Straßenkante wird bei diesem Kartenmaterial einer bestimmten Kategorie zugeordnet [Dam 2010, S. 7]:

*”Functional Class = 1 high speed and volume controlled access roads*

*Functional Class = 2 quick travel between and through cities*

*Functional Class = 3 moderate speed travel within cities*

*Functional Class = 4 moderate speed travel between neighborhoods*

*Functional Class = 5 lower speed travel within neighborhoods”*

Eliot Dam, NAVTEQ Director Product Management Latin America

In Abbildung 3.1 wird eine DLR Applikation dargestellt, die intern auch als ”NAVTEQ-Viewer” bezeichnet wird. Das Programm stellt die NAVTEQ Karte und das Netz dar. Exemplarisch wurde eine Route von der Autobahnauffahrt Adlershof bis zu der Autobahnauffahrt Neukölln in Berlin gewählt, um die hinterlegten Kanten aufzuzeigen. Bei jeder Kante wird die Fahrtrichtung als dreieckiger Pfeil angegeben.



**Abbildung 3.1.:** Screenshot der DLR Applikation ”NAVTEQ-Viewer”

<sup>6</sup>Link zum Unternehmen: <http://www.navteq.com/> (Zugriff: August 2011)

#### Darstellung der Verkehrslage

Bei dem FCD-System wird die Reisegeschwindigkeit - wie bereits in Kapitel 2.4.2 erläutert - auf Kanten abgebildet. Doch wie kann nun festgestellt werden, ob diese Reisegeschwindigkeit vielleicht aus einem Stau oder stockendem Verkehr bezogen wurde? Für eine klassifizierte Kante wird anhand der Straßenverkehrs-Ordnung (StVO)<sup>7</sup> die zulässige Geschwindigkeit bestimmt. Beispielsweise gehört eine Kante der Klasse 3 zu Straßen innerhalb geschlossener Ortschaften (z.B. Bundesstraßen, Fernstraßen). Laut §3 der StVO beträgt die zulässige Höchstgeschwindigkeit hier für Kraftfahrzeuge  $50 \frac{km}{h}$ . Die Prognose der Verkehrssituation wird durch die aktuelle Reisegeschwindigkeit auf einer Kante im Verhältnis zur zulässigen Geschwindigkeit ermöglicht. Das Verhältnis ( $p$  in Prozent) beider Geschwindigkeiten wird auf folgende Weise berechnet:

$$p = \frac{\text{aktuelle Geschwindigkeit}}{\text{zulässige Geschwindigkeit}}$$

Die Bestimmung der Verkehrssituation und die repräsentative Einfärbung einer Kante erfolgt nach folgenden Vorgaben:

- $p \geq 50\%$  - fließender Verkehr, die Kante wird grün eingefärbt
- $50\% > p \geq 25\%$  - stockender Verkehr, die Kante wird gelb dargestellt
- $p < 25\%$  - gestauter Verkehr, die Kante wird rot oder im Farbfehlsichtigkeits-Modus<sup>8</sup> blau eingefärbt

Ein Mapfile<sup>9</sup> umfasst beim DLR die Einfärbung von Kanten nach genannten Vorgaben für einen ganzen Ausschnitt (eine Stadt) und stellt sie als Schicht (eng. Layer)<sup>10</sup> bereit. In einer Webanwendung namens Cityrouter<sup>11</sup> wird die Verkehrslage bereits vom Institut für Verkehrssystemtechnik eingebunden und dargestellt. Ein Ausschnitt dieser Anwendung wird in Abbildung 3.2 gezeigt. Diese Verkehrslage

---

<sup>7</sup>Eine Referenz im Internet: <http://www.gesetze-im-internet.de/stvo/> (Zugriff: August 2011)

<sup>8</sup>Weitere Erläuterung hierzu folgt in Kapitel 4.1.3.

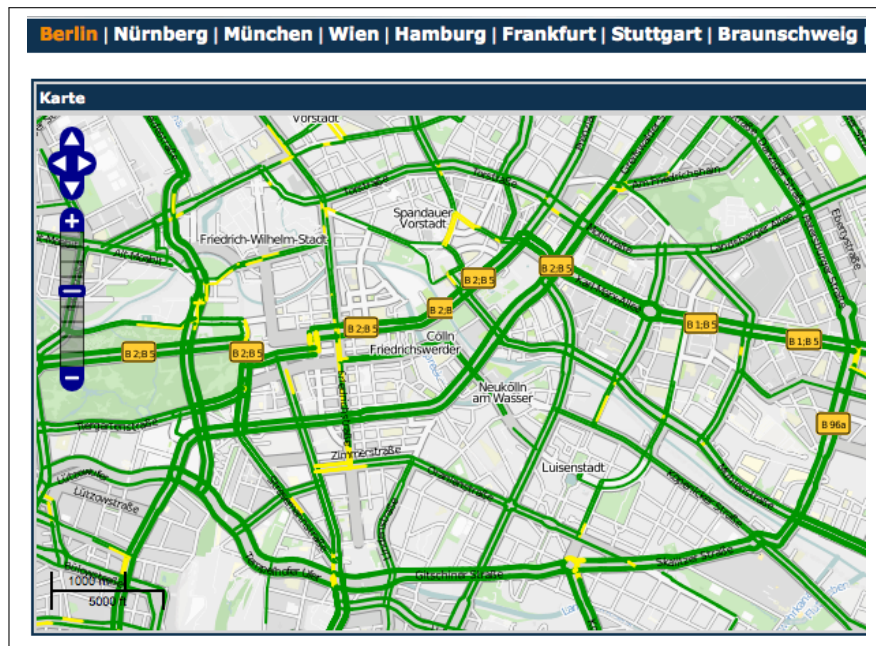
<sup>9</sup>Website zu der Konstruktbeschreibung des Anbieters: <http://mapserver.org/mapfile/> (Zugriff: August 2011)

<sup>10</sup>Ein Layer ist eine Schicht zum Anzeigen von Bildern. Entnommen aus der Beschreibung: <http://mapserver.org/mapfile/layer.html> (Zugriff: August 2011)

<sup>11</sup><http://cityrouter.de/cityrouter/cityrouter.html> (Zugriff: September 2011) Zugriff ist bis dato nur mit internen Nutzernamen und Passwort möglich.



### 3.1. Ist- und Sollzustand



**Abbildung 3.2.:** Screenshot der Cityrouter Website (04.09.2011 12:30 Uhr)

wird über ein Fast Common Gateway Interface (FastCGI)<sup>12</sup> als grafisches Objekt für einen MapServer<sup>13</sup> zur Verfügung gestellt und anhand der aktuellen Verkehrsdaten kontinuierlich neu erstellt. FastCGI erweitert CGI zusätzlich unter anderem durch eine Unterstützung von verteilten Systemen. So kann das FastCGI Skript in verteilten System auf unterschiedlichen Rechnern ausgeführt werden. Der MapServer ist eine Open-Source Programmeinheit zum Rendern von geografischen Daten und ermöglicht es, diese Daten als Kartenbilder darzustellen. Ein Renderer ist eine Funktionseinheit, die Kartenmaterial in Bilddateien umformt und erstellt. Eine Bilddatei wird in diesem Zusammenhang auch als Kachel bezeichnet.

#### OpenStreet Map Landkartenmaterial

Für die visuelle Darstellung von Kartendaten wird OSM verwendet. Es bietet im Vergleich zu NAVTEQ eine höhere Anzahl von verschiedenen Darstellungs- und Ein-

<sup>12</sup>Internetseite des Anbieters: <http://www.fastcgi.com> (Zugriff: August 2011)

<sup>13</sup>Internetpräsenz des Anbieters: <http://mapserver.org> (Zugriff: August 2011)

### 3.1. Ist- und Sollzustand

---

stellungsmöglichkeiten. Des Weiteren besteht mit OpenLayers<sup>14</sup> eine umfangreiche API, um unter anderem OSM in Webseiten einzubinden. Es ist eine auf JavaScript basierende Open-Source API für die Darstellung von Kartendaten und verfügt über viele weitere Features und Funktionen. Das OSM-Material wird beim Institut für Verkehrssystemtechnik speziell nach eigenen Einstellungen und Vorstellungen gerendert. Das Unternehmen erstellt die Kacheln für die Textdaten (unter anderem Stadt- und Straßennamen sowie Autobahnschilder) und Landkartendaten separat. Die kreierten Bilddateien werden auf einem Server des DLR zur Verfügung gestellt.

#### 3.1.2. Sollzustand

Es wird eine iPhone App prototypisch entwickelt, die im Wesentlichen folgenden Anforderungen entspricht: Bei der Programmierung besteht die Vorgabe darin, OpenStreet Map als Kartenmaterial einzusetzen und für die Handhabung und Steuerung der Karte OpenLayers zu verwenden. Diese webbasierte API (OpenLayers) muss für die Touchfunktionalität optimiert werden, wie zum Beispiel das Navigieren durch Ziehen auf der Karte. Zusätzlich zum geografischen Material und Straßennetz wird von Berlin die Verkehrslage mit Rot für gestaut, Gelb für stockend und Grün für fließend, auf dem Straßennetz angezeigt. Die Verkehrslage wird anhand von Verkehrsinformationen durch eine bereits bestehende Anwendung des DLR e.V. kontinuierlich generiert. Eine Positionsbestimmung des Smartphone-Nutzers wird sofort beim Start durchgeführt und die Position in der Karte angezeigt. Eine weitere Funktionalität wird der Vorschlag einer verkehrsoptimierten Route in Berlin sein, die anhand der aktuellen Position oder einer vom Nutzer vorgegebenen Startposition zu einer gewählten Zieladresse gebildet wird. Hierzu kann auf eine bereits bestehende Routing-Engine des DLR zurückgegriffen werden. Die Überwachung von personalisierten Routen, die zu einem bestimmten Zeitpunkt von einem Anwender als Route festgelegt wird, soll anhand von in der Datenbank hinterlegten Beispiel-Werten geschehen. Die Übertragungen von Positionsdaten des iPhone Nutzers als FCD werden an die Server Applikation in bestimmten Intervallen durchgeführt.

---

<sup>14</sup>Internetpräsenz von OpenLayers: <http://openlayers.org> (Zugriff: August 2011)

## 3.2. Anwendungsfallanalyse (Use Case)

### 3.2.1. Aus Sicht des Nutzers

Herr Müller ist seit mehreren Jahren Pendler in Berlin und kennt unterschiedliche Wege um sein Ziel zu erreichen. Er wählt seine Routen in Gebieten, die ihm vertraut sind, erfahrungsgemäß nicht so, wie sein Navigationsgerät es vorschlägt. Das hat bei ihm individuelle Gründe, wie die Bevorzugung von Nebenstraßen, Stauumfahrung oder Minimierung kritischer Abbiegevorgänge. Manchmal ist er auch der Meinung, dass auf der Straße, die sein Navigationsgerät empfiehlt, oft Ampeln mit langen Rotphasen seien und somit kein fließender Verkehr herrscht. Herr Müller möchte gerne von seinen ihm bekannten alternativen Strecken eine Route anhand der aktuellen Verkehrslage auswählen. Dank der Verkehrsinformations-App des DLR kann er sich auf seinem Smartphone schnell einen Überblick über die genaue Verkehrslage verschaffen. Durch die Routenüberwachung ist es ihm sogar möglich, schnell verkehrsoptimiert anhand der aktuellen Reisezeiten - im Vergleich zu historischen Verkehrssituationen auf der gespeicherten Route - eine Alternative auszuwählen. Die selektierte Route wird daraufhin in der Karte angezeigt. Nun kann Herr Müller losfahren und die Routenüberwachung parallel in der Karte anzeigen lassen. Aufgrund der Verkehrsinformationsapplikation und deren dynamischer Überwachung umgeht er effektiv stockenden oder gestauten Straßenabschnitten.

### 3.2.2. Aus Sicht des Unternehmens

Das Institut für Verkehrssystemtechnik des DLR e.V. sammelt, verarbeitet und prozessiert FCD, um Verkehrssituationen möglichst genau zu analysieren. Als Datenquellen wirken Taxiunternehmen bei einem vorhandenen Kooperationsvertrag. Nur in Gebieten, wo das DLR die Daten einer Taxiflotte erhält, kann eine Verkehrslage evaluiert werden. Zudem impliziert die Verkehrsanalyse nur Straßen, die von Taxifahrern befahren werden.

Durch die innovative und attraktive mobile Applikation werden Privatpersonen erreicht. Jeder neue Anwender ist von hoher Bedeutung für das DLR, denn die Bewegungsdaten des Smartphone-Nutzers werden an das FCD-System gesendet. Mit steigender Teilnehmerzahl wird die Akquisition von Verkehrsinformationen auf neuen Gebieten und Straßenabschnitten vorangetrieben und die Verkehrsprognosen wer-

### 3.2. Anwendungsfallanalyse (Use Case)

---

den zunehmend genauer. Über die Kooperationspartner hinaus können nun weitaus mehr Teilnehmer und somit "FCD-Quellen" gewonnen werden. Die fortschrittlichen Informationen ermöglichen dem DLR die Verkehrssituationen äußerst präzise und weit verbreitet zu erkennen. Dank der mobilen Applikation expandiert das FCD-System und unterliegt einem stetigen Wachstum. Es befähigt das Unternehmen, einen Vorsprung gegenüber der Konkurrenz auszubauen. In Anbetracht des neuen Datenbestands und dem bereits seit über 10 Jahren aufgebauten Know-how bei der FCD-Technologie, erscheint das Unternehmen und die Leistung zunehmend attraktiver. Bisher hat das DLR beispielsweise der Taxizentrale Braunschweig oder der VMZ Berlin Verkehrsdaten gegen Entgelt zur Verfügung gestellt. Aufgrund der mobilen Applikation wächst das Potenzial, weitere Kunden zu akquirieren und somit höhere Einnahmen zu erzielen.

## 3.3. Alternative Verkehrsinformationsapplikationen für Smartphones

### 3.3.1. Google

Google bietet seit dem 14. Juli 2011 Verkehrslagedaten in Deutschland an [Presstext.Redaktion 14. Juli 2011]. Die Informationen können zusätzlich zu den Karten in Google Maps angezeigt werden. Gegenüber der Nachrichtenagentur Presstext<sup>15</sup> gab ein Google Sprecher bekannt, dass die Verkehrsinformationen von Drittanbietern, Smartphones und Straßensensoren stammen [Presstext.Redaktion 14. Juli 2011]. Es werden Daten von Android und Google Navigation Nutzern anonym erhoben und ausgewertet [Barth 25. August 2009], um auf diese Weise eine Verkehrslage zu evaluieren. Die mobile Applikation "Karten"<sup>16</sup> ist eine vorinstallierte Applikation bei iOS. Sie beinhaltet zusätzlich zum Google Kartenmaterial die Möglichkeit, den Verkehr anzuzeigen und Routen zu wählen. Die Applikation verfügt nicht über eine Routenüberwachungs-Funktionalität. Bei Auswahl einer Route, welche Verkehrsstörungen beinhaltet, wird zusätzlich zur Reisezeit und -länge lediglich die Information "Bei Verkehr" bereitgestellt. Die genaue Position von Verkehrsstörungen wird an dieser Stelle den Nutzern nicht aufgezeigt. Des Weiteren bietet die App nicht die Möglichkeit, Karten offline zu betrachten. Bei einer Online-Nutzung werden Kartendaten in einen Offline-Cache abgelegt, nur diese Daten können bei einer Offline-Nutzung verwendet werden.

### 3.3.2. Navigon

Mit der mobilen Applikation namens "traffic4all"<sup>17</sup> bietet Navigon<sup>18</sup>, ein Unternehmen für Navigationslösungen, eine kostenlose Verkehrsinformationsapplikation für iPad und iPhone an. Zusätzlich zu einer angezeigten Verkehrslage für Europa können Nutzer Meldungen, wie Stau- oder Baustellen-Warnungen generieren und anderen Applikationsnutzern mitteilen. Die Verkehrsinformationen für eine Verkehrslage be-

---

<sup>15</sup>Website der Nachrichtenagentur: <http://www.pressetext.com> (Zugriff: August 2011)

<sup>16</sup>Link zu der App-Beschreibung von Apple: <http://www.apple.com/de/iphone/features/maps-compass.html> (Zugriff: August 2011)

<sup>17</sup>Link zu der Anwendung im App Store: <http://itunes.apple.com/de/app/navigon-traffic4all/id427793595> (Zugriff: August 2011)

<sup>18</sup>Internetpräsenz des Unternehmens: <http://www.navigon.com/> (Zugriff: August 2011)

### 3.3. Alternative Verkehrsinformationsapplikationen für Smartphones

---

zieht Navigon vom Kooperationspartner Inrix<sup>19</sup> [Pocketnavigation 21. April 2011]. In jedem Fenster der Applikation befindet sich ein großer Werbebanner, eine werbefreie Version ist mit einem Upgrade und damit verbundenen Kosten von 1,59 Euro erhältlich. Laut Kundenrezensionen wird der Werbebanner als störend, von Einigen sogar als Unverschämtheit und Grund zum Löschen der App empfunden. Begründet wird diese Empfindung in der Rezension damit, dass der Kunde dem Unternehmen Verkehrsinformationen bereitstellt, also nützliche und wertvolle Informationen liefert, aber für eine werbefreie Version zahlen muss. Die App verfügt über keine Routing- oder Routenüberwachungs-Funktionalität. Für eine Routing- und Navigation-Eigenschaft wird der iOS Nutzer auf die kostenpflichtige Smartphone Lösung von Navigon verwiesen, die für Europa und einem Preis von 89,99 Euro<sup>20</sup> im App Store erhältlich ist.

---

<sup>19</sup>Internetpräsenz des Unternehmens: <http://www.inrix.com/> (Zugriff: August 2011)

<sup>20</sup>Link zu der Navigon Navigationslösung im App Store: <http://itunes.apple.com/de/app/navigon-mobilnavigator-europe/id320279293> (Zugriff: August 2011)

## 3.4. Anforderungsanalyse

Um eine Analyse von Anforderungen (eng. Requirements) zu erleichtern, empfiehlt es sich die Requirements laut IEEE 1233 Standard [IEEE1233 1998] nach bestimmten Kriterien zu kategorisieren. Es werden folgende drei wesentliche Kategorien aufgestellt:

- *Funktionalität* - Einbindung und Steuerung von OpenStreet Map, GPS Unterstützung und alternative Ortungsmöglichkeiten, Kommunikationsschnittstellen (z.B. SOAP oder XHR)
- *Technische Unterstützung (Framework)* - standardisierte Programmiersprache, Dokumentation des Frameworks und APIs, Lizenz des Frameworks, Vertriebskanal vom Endprodukt (zum Beispiel App Store)
- *Qualität* - UI Darstellung des OS, Animationen

Nach IEEE 1233 ist es wichtig, dass die Kategorien und dazu spezifizierte Anforderungen messbar sind, um eine erfolgreiche Durchführung nachweisen zu können. Des Weiteren können unterschiedliche Prioritäten vorliegen, die dafür ausschlaggebend sind, welche Requirements Vorrang haben.

### 3.4.1. Anforderungsmatrix

Die selektierten und kategorisierten Anforderungen werden im nächsten Schritt den im Kapitel 2.3 vorgestellten Umsetzungsmöglichkeiten gegenübergestellt. Zur Visualisierung wird eine Anforderungsmatrix kreiert, in der zu den Anforderungen auch die Fähigkeiten der bestimmten Lösungsalternativen aufgezeigt und grundsätzlich bewertet werden. Die Anforderungsmatrix in Abbildung 3.3 dient als Grundlage für die weitere Analyse und die detaillierte Bewertung in Kapitel 3.5.

### 3.4. Anforderungsanalyse

Anforderungen		Umsetzung	Web-App	Native App	Hybride App		
					Titanium	Phonegap	Rhodes
Funktionalität	OpenStreet Map Einbindung		OpenLayers Javascript	zusätzliche native OSM Komponente nötig	OpenLayers Javascript	OpenLayers Javascript	OpenLayers Javascript
	Kommunikationsschnittstellen		Zusätzliche Programmkomponenten werden benötigt	einige Schnittstellen, Plug-Ins und Dokumentation	Beispiel SOAP Clientprogramm gegeben	zusätzliche Plug-Ins werden benötigt	Anwendungsbeispiele und Unterstützung
	GPS Ortung		n.V.	Hardware nah	Guter Zugriff, iOS4 Anfrage mit App Name	IOS4 location Anfrage unseriöses appname.app/www/index.html	Guter Zugriff, iOS4 Anfrage mit App Name
	Alternative Ortung		WIFI und IP Auswertungen über einen Google Service.	WLAN, Mobilfunk, digitaler Kompass	WLAN, Mobilfunk, digitaler Kompass	WLAN, Mobilfunk, digitaler Kompass	WLAN, Mobilfunk, digitaler Kompass
	Debugging		Browser Debugger	viele Möglichkeiten	Logging, Browser Debugger	Logging, Browser Debugger	Logging, Debugging
Technische Unterstützung	Programmiersprache		Websprache wie JavaScript, HTML und CSS	IOS -> Objektiv C Android= Java	JavaScript	Websprache wie JavaScript, HTML und CSS	Ruby und HTML
	Dokumentation		In Abhängigkeit von gewählten Frameworks	ausführlich und viele Beispiele	API dokumentiert, Beispiele als Video gegen Zahlung	API Dokumentation	API dokumentiert, gewerblicher Support nur gegen Zahlung
	Lizenz		verschiedene Open-Source Lizenzen	verschiedene Lizenzen: Apple-EULA, GPL, APSL u.a.	Open-Source Apache	Open-Source MIT	Open-Source MIT, Abhängigkeit von kostenpflichtigen Zusatzprodukten
	Plattformübergreifend		viele	n.V.	IOS, Android, BlackBerry Beta	viele	viele
	Vertrieb		Kein Vertrieb im App Store. Auf Apple Seite aber anbietbar	Nach gewählter nativer Sprache nur für diesen App Markt geeignet	Laut Anbieter sowohl Google Market als auch Apple Store	Laut Anbieter gleicher Vertrieb wie bei nativem Produkt	Laut Anbieter gleicher Vertrieb wie bei nativem Produkt
Qualität	UI Darstellung		Dritt-Anbieter wie jQueryTouch	Optimale Unterstützung	Viele native Komponenten	Dritt Anbieter wie jQueryTouch	Viele OS Komponenten
	Animationen		Dritt-Anbieter wie jQueryTouch	Optimale Unterstützung	viele OS typische	Dritt-Anbieter wie jQueryTouch	viele OS typische

Legende
Gut
Mittel
Schlecht

Abbildung 3.3.: Anforderungsmatrix



## 3.5. Analyse der Bewertungsmatrix

Anhand der Anforderungen und deren Priorität sowie Gewichtung werden entscheidende Faktoren zur Bewertung der Umsetzungsmöglichkeit gegeben. Hierdurch kann eine Lösungsvariante ausgewählt werden. Im ersten Schritt wird zu jeder Anforderung die Umsetzungsmöglichkeit einer Variante durch die Vergabe von Punkten (0-5 Punkte) bewertet. Die Vergabe der höchsten Punktzahl bedeutet, dass die Anforderung durch die Variante sehr gut umgesetzt werden kann. Im nächsten Schritt wird für jede Spalte (Umsetzungsmöglichkeit) eine Summe berechnet, indem für jede Zeile der Spalte ein Produkt berechnet wird und diese Produkte addiert werden. Das Produkt ergibt sich durch die Multiplikation der Bewertung mit der gewichteten Priorität (1-10 Punkte) einer Anforderung. Je höher die Summe ist, desto qualifizierter ist die Umsetzungsvariante als Lösung in Bezug auf die Anforderungen und ihre Gewichtung. Die Bewertungsmatrix wird in Abbildung 3.4 gezeigt.

### 3.5.1. Auswahl des Lösungsweges

Aus der Bewertungsmatrix geht hervor, dass die Entwicklung einer hybriden App mit dem Framework Titanium Appcelerator anzustreben ist. Bezüglich der aufgestellten Anforderungen zeigte das Framework im Vergleich zu anderen Umsetzungsmöglichkeiten vorteilhafte Fähigkeiten auf. Aufgrund dessen wird im Kapitel 4 ein Konzept und eine Implementierung mit Titanium Appcelerator indiziert.

### 3.5. Analyse der Bewertungsmatrix

Legende

Gut

Mittel

Schlecht

Umsetzung		Priorität	Web-App	Native App	Hybride App		
Anforderungen					Titanium	Phonegap	Rhodes
Funktionalität	OpenStreet Map Einbindung	9	5	3	5	5	5
	Kommunikationsschnittstellen	7	3	5	4	3	4
	GPS Ortung	10	0	5	5	3	5
	Alternativ Ortung	8	3	5	5	4	5
	Debugging	8	3	5	3	3	3
Technische Unterstützung	Programmiersprache	9	5	1	4	5	2
	Dokumentation	6	3	5	3	3	3
	Lizenz	9	5	5	5	5	1
	Plattformübergreifend	10	5	0	4	5	5
	Vertrieb	10	2	2	5	5	5
Qualität	UI Darstellung	9	2	5	4	2	4
	Animationen	6	2	5	4	2	4
Summe			322	371	436	390	392

Abbildung 3.4.: Bewertungsmatrix

---

## 4. Konzept und Umsetzung der mobilen Applikation

In Kapitel 4.1 wird das Konzept der mobilen Applikation erarbeitet. Das nachfolgende Kapitel 4.2 umfasst wesentliche Informationen und Schritte, die bei einer Entwicklung mit Titanium Appcelerator für das iPhone unabdingbar sind. Hierzu zählt die Installation von benötigten Programmen. Weiterhin wird ein Einstieg in das Framework gegeben. Auf Basis des Konzeptes wird in Kapitel 4.3 die Implementierung der mobilen App beschrieben und anhand von bildlichen Programmausschnitten exemplifiziert.

### 4.1. Konzept

Die mobile Applikation wird hinsichtlich der Einflussfaktoren von Usability, Accessibility, Manual, Design, Human Interface und Interaction einer User Experience (siehe Kapitel 2.2) konzipiert. Das Konzept wird speziell danach ausgerichtet, auf welche Weise bei der Anwendung ein positives Nutzererlebnis geschaffen wird. Dafür werden unter anderem Funktionen und Dienste genannt, die einen echten Mehrwert aufzeigen und die Applikation attraktiv und innovativ erscheinen lassen. Um das vollständige Potenzial der App zu offenbaren und konzeptionell zu benennen, ist es wichtig, über den Prototyp hinaus mögliche zukünftige Eigenschaften und Funktionen der Anwendung nahezulegen. Die zukünftigen Betrachtungen werden in diesem Kapitel *kursiv* gekennzeichnet.

### 4.1.1. Brauchbarkeit und Nutzbarkeit

Die Brauchbarkeit und Nutzbarkeit (eng. Usability) der Applikation wird im Folgenden als Vereinigung signifikanter Eigenschaften von Effizienz, Effektivität, Zuverlässigkeit und Funktionalität konzipiert. Die Effektivität beschreibt die Wirksamkeit und Praktik sowie persönliche Effektivität (persönlicher Vorteil des Nutzers durch die App). Mit Effizienz werden sowohl die Performance der Software, als auch die Leistungsfähigkeit bestimmter Applikationsaufgaben beschrieben. Die Zuverlässigkeit umfasst dabei die Aspekte des Datenschutzes, Vertrauens von Nutzern (wodurch das Vertrauen gefördert wird), die Sicherheit und Fehlerbehandlungen. Mittels der Funktionalität werden die Dienste der Applikation aufgezeigt und vor allem innovative Funktionen genannt.

### Funktionalität

Der Prototyp impliziert die Funktionalitäten der Positionsbestimmung, die Erfassung und periodische Übertragung von FCD zum DLR, Kommunikation mit dem Server, Karten- und Verkehrslagedarstellung, verkehrsoptimiertes Routing und Überwachung von personenbezogenen Strecken.

- **Kommunikation**

Die Kommunikation zwischen Client und Server wird anhand eines einheitlichen Protokolls geschehen, das einer Standardisierung zugrunde liegt. Serverseitig wird ein Web Service entwickelt, der über das Simple Object Access Protocol (SOAP) mit dem Client kommuniziert. SOAP ist eine Spezifikation des World Wide Web Consortiums (W3C)<sup>1</sup> und basiert auf Extensible Markup Language (XML)<sup>2</sup>. Weitere Beschreibungen des Web Services folgen in Kapitel 5.2.

Für Datenobjekte wird als Übertragungsformat JavaScript Object Notation (JSON)<sup>3</sup> gewählt. Es ist ein auf JavaScript basierendes kompaktes Datenformat und im Vergleich zu XML kann es Objekte mit einer geringeren Anzahl von Zeichen beschreiben [Schmidt 17. Juli 2009, S. 50-51].

---

<sup>1</sup>Link zur Spezifikation von SOAP Version 1.2 <http://www.w3.org/TR/soap12-part1> (Zugriff: August 2011)

<sup>2</sup>Spezifikation des WC3: <http://www.w3.org/TR/xml/> (Zugriff: August 2011)

<sup>3</sup>Internetpräsenz des Formates: <http://www.json.org/> (Zugriff: August 2011)

## 4.1. Konzept

---

- **Positionierung des Gerätes, je nach Vorhandensein mit GPS, Mobilfunk oder WLAN**

Eine Lokalisierung kann erfolgen, auch wenn zu dem Zeitpunkt der Ortung kein Satellitenempfang des GPS vorhanden ist. Hierfür muss im mobilen Endgerät entweder Mobilfunk oder Wi-Fi aktiviert sein und ein entsprechender Empfang vorliegen. In Kapitel 3.5.1 wurde Titanium als fähigste Umsetzungsmöglichkeit bewertet und ausgewählt. Alle Positionsinformationen und Funktionen, die iOS durch die Klasse "CLLocationManager" bereitstellt (siehe Kapitel 2.5.4), werden von Titanium durch das Modul "Titanium.Geolocation"<sup>4</sup> zur Verfügung gestellt. Somit wird die Positionsermittlung über die API von Titanium durchgeführt. Ebenso stellt das Modul Funktionen bereit, um vor erstmaliger Lokalisierung die ab iOS 4 benötigte Berechtigung zu erfragen. Erlaubt der Nutzer keine Ortungsinformationen für die App, müssen entsprechende Empfehlungen und Informationen angezeigt werden. Dieses Feedback wird in Kapitel 4.1.2 weiter konzipiert und erläutert. Die Positionsermittlung wird kontinuierlich durchgeführt, um das Zeichnen der aktuellen Position in der Karte und das periodische Sammeln von FCD zu unterstützen.

- **Karten- und Verkehrslagedarstellung**

Das Kartenmaterial von OpenStreet Map wird als Webview in der Titanium Applikation mit OpenLayers eingebunden. Eine mobile API von OpenLayers wird unter dem Dateinamen "OpenLayers.mobile.js" in der HTML-Datei "mapwidget.html" eingesetzt. Beide Dateien werden im Projekt gespeichert. Auf diese Weise müssen die Daten und Skripts vor einer Nutzung nicht zuerst über das Internet bezogen werden. "OpenLayers.mobile.js" ist eine kompakte und verkleinerte Version der OpenLayers API, die für mobile Endgeräte wesentliche Objekte und Funktionen beinhaltet. Nicht in der kompakten Version enthaltene Komponenten müssen von der vollständigen API<sup>5</sup> bezogen und intern für die App gespeichert werden. Innerhalb dieser Ausarbeitung werden folgende Komponenten zusätzlich von der unverkürzten API bezogen:

- "Markers.js" - Das Objekt bietet die Möglichkeit, Bilder an geografischen

---

<sup>4</sup>Link zur Dokumentation des Moduls: <http://developer.appcelerator.com/apidoc/mobile/1.7.1/Titanium.Geolocation-module> (Zugriff: August 2011)

<sup>5</sup>Link zu dem OpenLayers Projekt: <http://trac.osgeo.org/openlayers/browser/trunk/openlayers/lib/OpenLayers> (Zugriff: August 2011)

## 4.1. Konzept

---

Orten zu zeichnen. Die Funktionseinheit wird zum Zeichnen der Position auf der Karte eingesetzt.

- "MapServer.js" - Wie in Kapitel 3.1.1 bereits beschrieben, wird die Verkehrslage für einen MapServer zur Verfügung gestellt. OpenLayers bietet in der API mit "MapServer.js" einen MapServer an. Dieses Konstrukt ermöglicht den Bezug und die Darstellung der Verkehrslage.

Im Zusammenhang mit der Einbindung von Kartendaten wird für die Webview ein großer Bereich in der Applikation benötigt. Um dem Anwender einen übersichtlichen und informationsreichen Kartenausschnitt zu Verfügung zu stellen, ist eine möglichst große Darstellungsfläche wichtig. Die Webview wird von Kartenbildern komplett ausgefüllt und auf diese Weise der verfügbare Platz vollkommen ausgeschöpft.

Im Istzustand (Kapitel 3.1.1) wurde beschrieben, dass die Karten- und Textbilder getrennt vorliegen. OpenLayers bietet mit "OpenLayers.Layer.OSM" eine Instanz an, um OpenStreet Map Kacheln als Kartenmaterial einzubinden. Das Modul basiert auf "OpenLayers.Layer.XYZ"<sup>6</sup>. Es bietet die Möglichkeit, Kacheln in einem Gitter zu laden. Jede Kacheldatei ist im Gitter durch x, y (Zeile, und Spalte) sowie z (Zoomstufe) eindeutig identifizierbar. Aufgrund dieser Umsetzung müssen nicht alle Kacheln übertragen werden, sondern nur spezifische, die für einen betrachteten Ausschnitt zutreffen. Eine Veränderung der Ansicht<sup>7</sup> bewirkt automatisch ein Nachladen von benötigten Kacheln.

Es wird bei der Implementierung für die Karten- und Textkacheln jeweils ein OSM Objekt erstellt. *In Zukunft werden die Kartendaten offline auf dem Endgerät hinterlegt sein. Somit müssen die Bilddaten nicht während der Nutzung übertragen werden. Eine Aktualisierung von offline hinterlegten Karten (zum Beispiel bei Veränderung des Straßennetzes) wird ebenfalls möglich sein.*

Weiterhin werden folgende Funktionalitäten im Zusammenhang mit der Kartenkomponente offeriert:

- Vollbildmodus - Wird der Vollbildmodus aktiviert, erscheint die Karte alleine und vollständig auf dem Display, indem die Tab- und Navigati-

---

<sup>6</sup>Link zur Quelldatei des Konstrukts: <http://trac.osgeo.org/openlayers/browser/trunk/openlayers/lib/OpenLayers/Layer/XYZ.js> (Zugriff: August 2011)

<sup>7</sup>Zum Beispiel, wenn die Zoomstufe verändert oder auf der Karte zu einem anderen Punkt gezogen wurde.

## 4.1. Konzept

---

onsbar ausgeblendet werden. Der Anwender kann auf diese Weise einen größeren Kartenausschnitt auf dem Display betrachten.

- Lupenmodus - Standardmäßig wird der aktuelle Aufenthalt als grafischer Punkt automatisch auf der Karte zentriert dargestellt. Ist der Lupenmodus aktiviert, kann der Nutzer die Kartenumgebung anschauen, ohne dass eine automatische Zentrierung der Position stattfindet.
- *Auswahl von verschiedenen Kartenmaterialien - Zukünftig wird es möglich sein, verschiedene Kartendarstellungen zu wählen. Es kann zwischen einer Satellitenkarte oder farblich variierenden Karten gewählt werden.*

- **Nice-to-have<sup>8</sup>: Routen-Monitor**

In der Applikation können Routen gespeichert und gelöscht werden. Auf diese Weise können Nutzer persönliche Routen zusammenstellen. Festgelegte und gespeicherte Routen lassen sich parallel beobachten. Vorhandene Routen werden über die Kommunikationsschnittstelle abgefragt, indem der Web Service über SOAP aufgefordert wird, die für den Nutzer spezifischen Strecken auszuliefern. Dazu müssen die Nutzerdaten beim SOAP-Call mit übergeben werden. Jedes Routenobjekt wird als JSON Zeichenkette übertragen und kann somit auf Client-Seite verwendet werden. Anhand dieser Daten werden Informationen über die Reisezeit und Streckenlänge der Route bestimmt. Diese Daten werden dem Nutzer zusätzlich zum eingestellten Routennamen visualisiert. Somit kann der Anwender jederzeit einen Überblick über seine gespeicherten Routen gewinnen. Zusätzlich zur Länge und aktuellen Fahrzeit der Strecke werden auftretende Verzögerungen erkenntlich gemacht. Jede Route wird als waagerechte Linie gezeichnet und an zutreffenden Stellen wird die Verzögerungen durch senkrechte Striche sowie zeitliche Angaben eingezeichnet. *Zukünftig wird der Routen-Monitor grafisch verbessert. Die Verkehrslage wird farblich auf der Linie eingezeichnet. Zudem kann die Position des Nutzers auf der Linie dargestellt werden, wenn er die gespeicherte Route befährt.* Der Routen-Monitor ist eine innovative Funktionalität, da beliebige Routen überwacht und auch auf Nebenstraßen Verkehrsinformationen geliefert werden. Im Rahmen dieser Arbeit konnte keine alternative Applikation mit einer derartigen

---

<sup>8</sup>Keine Sollanforderung, aber es ist ein Feature, das sich lohnt zu haben.

## 4.1. Konzept

---

Funktionalität gefunden werden.

- **Routing**

In Bezug auf das Routen-Monitoring muss für den Zugriff auf die Routing Funktionalität die identische Kommunikationsschnittstelle zum Server verwendet und somit der Web Service für die mobile App angesprochen werden. Damit der Anwender den momentanen Aufenthalt beim Routing als Adresse verwenden kann, wird ein auswählbares Feld mit der Bezeichnung "Aktuelle Position" zur Verfügung gestellt. Zusätzlich zu dieser Bezeichnung wird die Adresse in dem selektierbaren Feld angezeigt. Die Adresse wird durch die vorliegenden Ortungsdaten vom Service bestimmt. Auf diese Weise ist der Nutzer informiert, welche Adresse als derzeitiger Aufenthalt verwendet wird. Die aktuelle Position kann entweder als Start- oder Zieladresse gewählt werden. Alternativ zu einer Verwendung der momentanen Position können in den Textfeldern Straßen eingegeben werden. Ist nach einer Eingabe von Ziel- und Startadresse eine Internetverbindung vorhanden, erfolgt eine Routenanfrage. Wenn die Eingaben korrekt waren und passende Straßen gefunden wurden, berechnet der Server eine Route und übermittelt diese der Applikation. Andernfalls wird der Nutzer entsprechend über Fehleingaben oder benötigte Eingaben informiert. Besteht bei einer Routensuche kein Internetempfang, ist ein Routing nicht möglich. In diesem Fall wird der Anwender ebenfalls benachrichtigt. *Optional wird in Zukunft eine Zwischenadresse für das Routing wählbar sein. Auf diese Weise können alternative Routen über bestimmte Zwischenhalte gebildet werden.*

- **Erfassung und periodische Sendung von FCD zum DLR Server**

Innerhalb der kontinuierlichen Ermittlung von Geopositionen werden die periodisch erfassten Daten gesammelt. Vor der Speicherung von Positionsinformationen als FCD findet eine Verifizierung der Daten statt. Qualitativ unbrauchbar sind Positionsinformationen, deren Genauigkeit von 13 Metern (maximale GPS Ungenauigkeit<sup>9</sup>) abweicht. Weiterhin werden Positionen verworfen, bei denen der Zeitabstand in Bezug auf bereits erhaltene Daten nicht mindestens 10 Sekunden beträgt. Eine Sammlung von Daten mit Zeitabständen unter 10

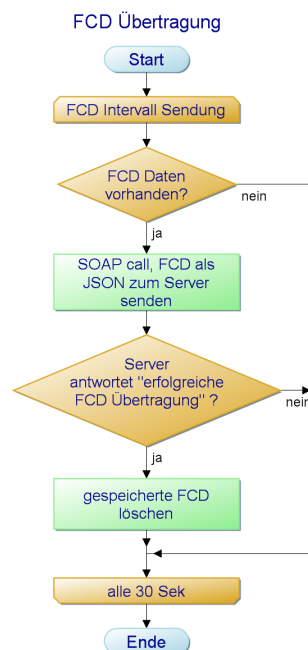
---

<sup>9</sup>siehe Kapitel 2.5.1



## 4.1. Konzept

Sekunden würde zu redundanten Informationen führen (siehe Kapitel 2.4.2). Die Festlegung des Intervalls für Sendungen wird bezüglich des in Kapitel 2.4.2 erläuterten FCD-Systems vollzogen. Als Zeitintervall werden 30 Sekunden festgelegt. Der Informationsgehalt von FCD ist demzufolge bei der App etwa 3 bis 6 Mal höher als der Datengehalt beim Taxi-System (hier wird eine Position zwischen 30 und 60 Sekunden gesendet). Die periodische Übertragung von FCD wird anhand eines Diagramms als Ablaufplan konzipiert. Dieser Ablaufplan wird in Abbildung 4.1 dargestellt.



**Abbildung 4.1.:** Periodische Sendung von FCD als Ablaufplandiagramm

Im Zusammenhang mit dem gezeigten Diagramm ist mit Start und Ende die Laufzeit der Applikation gemeint. Eine Beendigung der Übertragung erfolgt somit beim Schließen der Applikation. Wie im Diagramm abgebildet, wird alle 30 Sekunden folgender Prozess als Schleife durchgeführt:

- Zuerst wird überprüft, ob überhaupt Daten für eine Sendung vorliegen.
- Liegen keine Daten vor, werden die nächsten Vorgänge übersprungen.
- Sind Daten vorhanden, wird der Web Service per SOAP aufgerufen und

## 4.1. Konzept

---

FCD übertragen. FCD wird mittels JSON schlank gesendet.

- Der Server antwortet auf die Sendung.
- Handelt es sich um eine positive Antwort, hat die Übertragung ohne Fehler geklappt und die zuvor übertragenen Daten können gelöscht werden. Andernfalls erfolgt keine Löschung.

### Effizienz

Anstehende Aufgaben, wie ein Fensterwechsel (Tabwechsel), werden von der Applikation effizient umgesetzt. Eine sofortige Aussagefähigkeit ist von hoher Wichtigkeit, damit ein Anwender nicht auf Ergebnisse warten muss. Die Ortung erfolgt direkt mit dem Applikationsstart, auch wenn zu dem Zeitpunkt kein GPS vorhanden ist. Diese Funktionalität wird in Kapitel 4.1.1 beschrieben. Nach einer Lokalisierung wird die Position sofort in der Karte angezeigt.

Für die Bestimmung einer Route wird die aktuelle Position in einer Liste vorgegeben. Diese ist schnell auszuwählen und somit als Ziel- oder Startpunkt anwendbar. *Zukünftig werden häufig verwendete und gesuchte Straßen in einer Liste vorgeschlagen, damit der Nutzer seine gewöhnlichen Anfragen angezeigt bekommt und diese sofort auswählen kann.* Bei einer Routing-Adresse reicht es, wenn der Nutzer eine Straße angibt, und selbst die muss bei einer Eingabe nicht vollständig sein. Beispielsweise werden dem Nutzer für die Eingabe der Zeichenkombination "Magnu" alle Straßen in Berlin zur Auswahl vorgeschlagen, die für diese Zeichenkombination infrage kommen. *In Zukunft kann die Stadt optional mit angegeben werden, um das Routing in vielen Regionen effizient zu ermöglichen und Verwechslungen von Straßen von vornherein auszuschließen.*

### Effektivität

Die Vorteile für einen Nutzer und das Unternehmen wurden mittels der Anwendungsfallanalyse in Kapitel 3.2 genannt. Die App ermöglicht Anwendern, im Verkehr schneller ein persönliches Verkehrsziel zu erreichen.

## 4.1. Konzept

---

### Zuverlässigkeit

Es besteht eine zuverlässige Arbeitsweise der Software. Unvorhersehbares Softwareverhalten (Absturz) wird in der Testphase bei Vorhandensein identifiziert und protokolliert. Das Vertrauen von Nutzern ist von hoher Wichtigkeit. Nach §3a des Bundesdatenschutzgesetzes (BDSG) ist, wenn es möglich ist und der Aufwand in einem angemessenen Verhältnis zum Schutzzweck steht, eine Anonymisierung<sup>10</sup> und Pseudonymisierung<sup>11</sup> anzuwenden. Der Nutzer wird über sicherheitsrelevante Aspekte, wie Datenschutz und Anonymisierung in geeigneter Weise informiert. Durch Transparenz wird das Vertrauen gefördert. Wie wird zum Beispiel die Verkehrslage gewonnen und wie werden Daten für die Verkehrsanalysen anonymisiert verwendet? *An geeigneter Stelle der Applikation gilt es, zukünftig den Anwender darüber zu informieren. Es bietet sich an, dies bei der ersten Ausführung der Anwendung zu vollziehen. Exemplarisch kann eine derartige Information inhaltlich wie folgt aussehen: "Es geschieht keine Datenweitergabe oder eine ungewollte Veröffentlichung von Positionsdaten. Die anonymisierten Ortungsinformationen (geografische Informationen und Geschwindigkeiten) werden von vielen Applikationsnutzern und Kooperationspartnern gesammelt und prozessiert. Dies geschieht alleinig zum Zweck, eine Verkehrslage zu evaluieren und genaue Verkehrsinformationen zur Verfügung stellen zu können. Jeder Nutzer trägt seinen Teil dazu bei, dass unser Verkehrsinformationsdienst noch genauer wird." Eine rechtswirksame Fassung wird von einer juristischen Fachkraft erstellt. Weiterhin werden auf diese Weise die Allgemeinen Geschäftsbedingungen (AGB) und Datenschutzerklärungen für die Anwendung aufgestellt. Demzufolge werden Richtlinien und Informationen für Datenschutz und Allgemeine Geschäftsbedingungen entworfen, die einem Applikationsnutzer über eine Informationsseite (siehe Kapitel 4.1.4) transparent nahegelegt werden. Des Weiteren werden zukünftig Konzepte für die Sicherheit der Applikation und Kommunikation entworfen.*

---

<sup>10</sup>Nach §3 Absatz 6 des BDSG: Die Veränderung von personenbezogenen Daten, damit Verhältnisse einer Person nicht mehr zugeordnet werden können oder die Zuordnung erschwert wird.

<sup>11</sup>Nach §3 Absatz 6a des BDSG: Das Ersetzen von Erkennungsmerkmalen einer Person durch eine Kennung, um die Bestimmung der Person zu verwehren oder zu erschweren.

## 4.1. Konzept

---

### 4.1.2. Interaktion

Die Interaktion (eng. Interaction) des Nutzers mit dem System wird durch Rückmeldung (eng. Feedback) sowie Dynamik und Flexibilität der Applikation gebildet. Das Feedback umfasst visuelle Rückmeldungen und Statusanzeigen. Mit der Dynamik und Flexibilität werden Fähigkeiten der Anwendung beschrieben, auf individuelle Bedürfnisse und Anforderungen des Nutzers aktiv reagieren zu können.

#### Rückmeldung

Der Anwender wird korrespondierend darüber informiert, ob eine Ortung erfolgen konnte und ob die Position genau oder nur grob evaluiert wurde. Zudem wird der Status angezeigt, ob eine Internetverbindung besteht. Es ist wichtig, dem Nutzer diese Informationen bereitzustellen, um die Interaktion zu fördern. Anwender können Feedback, Kritik und Bugs über eine angegebene Kommunikationsschnittstelle (E-Mail-Adresse) äußern.

Laut Kapitel 2.5.4 muss der Benutzer bei erster Ausführung der App seit iOS Version 4 einer Ortung durch das Programm zustimmen. Die Applikation wird entsprechend darauf reagieren, wenn eine Ortung vom Nutzer nicht akzeptiert wird. Es ist unabdingbar für eine FCD-Funktionalität, dass eine Ortung kontinuierlich durchgeführt werden kann. Somit muss der Nutzer darauf hingewiesen werden, dass eine Ortung zu aktivieren ist, um den vollen Funktionsumfang der Applikation nutzen zu können. Es wird gezielt darauf hingewiesen, dass ohne Ortung nur eine eingeschränkte Version möglich ist. Auf diese Weise wird dem Anwender klar dargelegt, dass die Applikation nur mit aktivierter Ortung sinnvoll nutzbar ist.

#### Dynamik und Flexibilität

Eine flexible Anwendung zeichnet sich darin aus, dass die Darstellung dynamisch veränderbar ist. Über die Einstellungen sind für den Nutzer während der Anwendung folgende Funktionen und Ansichten konfigurierbar:

- Fehlsichtigkeitsmodus - Nach Aktivierung oder Deaktivierung des Fehlsichtigkeitsmodus wird die Verkehrslage und deren Legende mit entsprechenden Farben dargestellt. Der Fehlsichtigkeitsmodus kann eingestellt werden und wird persistent gespeichert. Bei Neustarten der Applikation ist die Fehlsich-

## 4.1. Konzept

---

tigkeit somit sofort aktiviert und die Verkehrslage sowie die Legende werden entsprechend für den Nutzer dargestellt.

- Sichtbarkeit der Verkehrslage - Die Anzeige der Verkehrslage kann aktiviert oder deaktiviert werden
- Vollbildmodus der Karte - nach Aktivierung oder Deaktivierung der Vollbildoption wird die Karte entsprechend dargestellt. Mit dem Vollbildmodus wird dem Nutzer - auf individuellen Wunsch hin - ein deutlich größerer Kartenausschnitt ermöglicht.
- *Das Erscheinungsbild der Software kann von einem Nutzer modifiziert werden. Die Ordnung der Tabs lässt sich nach individuellem Wunsch verändern.*

### 4.1.3. Barrierefreiheit

Die Zugänglichkeit und Barrierefreiheit (eng. Accessibility) wird nach körperlicher und sprachlicher Einschränkung separiert.

#### Barrierefrei bei körperlichen Einschränkungen

Die App bietet für Nutzer mit einer Rot-Grün-Sehschwäche<sup>12</sup> die Möglichkeit eines Fehlsichtigkeits-Modus. Verwendete Farben werden bei der Visualisierung und Aktivierung des Modus entsprechend angepasst. Die Verkehrslage wird auf dem Straßennetz statt Rot für gestaut, Gelb für stockend und Grün für fließend, in den Farben Blau, Gelb und Grün dargestellt.

#### Barrierefrei bezüglich der Sprache

Der Prototyp unterstützt die deutsche Sprache. *Zukünftig wird eine englischsprachige Unterstützung folgen.*

### 4.1.4. Anleitung

Eine Anleitung (eng. Manual) zählt laut Donald Norman (siehe Kapitel 2.2) zu einem Faktor, der für eine positive User Experience wichtig ist. Als eine wesentliche

---

<sup>12</sup>Eine Unterscheidung von Rot und Grün ist nicht möglich.

## 4.1. Konzept

---

informierende Unterstützung wird die App über eine Informationsseite verfügen. Die Informationsseite beinhaltet aber keine Anleitung, wie die App zu steuern ist. Denn laut Apple (siehe 2.2.1) ist eine Applikation zu erstellen, die ohne Anleitung für einen Nutzer verständlich und anwendbar ist. Stattdessen wird dem Anwender über die Informationsseite Transparenz aufgezeigt. Zum einen wird das DLR als Service Anbieter mit Kontaktdaten vorgestellt. Ebenfalls wird hier ein Ansprechpartner genannt. *Zum anderen werden wichtige Informationen über Datenschutz und über die Allgemeinen Geschäftsbedingungen auf dieser Seite zum Nachschlagen hinterlegt.* Die Angabe eines Ansprechpartners zeigt eine Unterstützung auf, indem den Nutzern für Kritik, Anregungen oder Fragen eine Kommunikationsstelle offengelegt wird. Diese Informationsseite wird an einem zentralen Ort als ein Tab Fenster hinterlegt. Eine übersichtliche Visualisierung dieser Seite unterstützt den Anwender dabei, sich Klarheit zu verschaffen. *Zukünftig wird es eine Online-Hilfe und eine Übersicht mit häufig gestellten Fragen (FAQ) geben.*

### 4.1.5. Design

Das Design der Applikation wird durch Konsistenz und durch eine klare Anordnung (eng. Layout) attraktiv. Wie bereits in Kapitel 2.2.1 erläutert, werden die für Konsistenz typischen UI-Controlls, -Views oder -Icons und Standards verwendet. Diese gilt es, in einem richtigen Kontext einzusetzen. Eine klare Gestaltung wird durch Übersichtlichkeit und Organisation der Elemente sowie durch eine logische Anordnung erzielt.

### Klare Gestaltung der Benutzeroberfläche

Grundsätzlich werden in der App vier wesentliche Komponenten und Funktionen angeboten:

- Karte mit Verkehrslage
- Routing
- Routenüberwachung
- Einstellungen

## 4.1. Konzept

---

Für die Übersichtlichkeit und eine klare Definition gilt es, diese vier grundlegende Funktionen in der grafischen Benutzeroberfläche zu trennen. Zugunsten der iOS typischen Tabbar-Elemente werden diese in einem Hauptapplikationsfenster separat zur Verfügung gestellt. Auf diese Weise sieht der Anwender die wesentlichen und wichtigen Funktionen sofort. Durch das bekannte und verbreitete Tabbar-Konstrukt (z.B. wird Tabbar in Anwendungen wie iTunes, App Store und Media Player verwendet) ist die Handhabung bekannt und die Separierung der Komponenten geläufig.

### Konsistenz

Typische UI-Elemente oder -Icons werden eingesetzt und in einem richtigen Kontext verwendet. Exemplarisch wird das iOS typische Icon für eine "Refresh-Aktion" eingesetzt, um die Karte oder die überwachten Routen manuell aktualisieren zu lassen. Für Einstellungen werden die typischen Slider- und Check-Buttons verwendet, um bestimmte Aktionen zu aktivieren oder zu deaktivieren.

### 4.1.6. Benutzerschnittstelle

Für die Steuerung und Interaktion des Gerätes durch den Menschen (Benutzerschnittstelle, eng. Human-Interface) werden Multi-Touch Gestiken unterstützt und eingesetzt. Auf diese Weise wird eine höhere Affinität und Sensibilität von Nutzern für die Handhabung und Steuerung der App erzielt. Unter Verwendung von OpenLayers wird das webbasierte Kartenmaterial für die Touchfunktionalität optimiert, zum Beispiel: Navigieren durch Ziehen auf der Karte. Seit dem 25.02.2011<sup>13</sup> werden Touchfunktionalitäten durch OpenLayers sowie andere spezielle mobile Eigenschaften und Einstellungen ermöglicht. Dafür wird das Objekt "OpenLayers.Control.TouchNavigation" dem OpenLayers Map-Objekt als Control hinzugefügt. Daraufhin unterstützt das Framework typische iOS Gestiken (siehe Kapitel 2.2.2).

Wie bereits im Kapitel 2.2.1 thematisiert wurde, ist es von Bedeutung, eine gute Balance zwischen programmtechnischer Automatisierung und manueller Kontrolle zu finden. Der Nutzer soll die App möglichst eigenständig steuern können. Informationen über die Ortungsgenauigkeit oder Internetverfügbarkeit werden nach manu-

---

<sup>13</sup>Die Umsetzung erfolgte durch ein Team auf einer Konferenz: <http://openlayers.org/blog/2011/02/25/mobile-sprint-day-five/> (Zugriff: August 2011)

#### 4.1. Konzept

---

eller Auswahl angezeigt. Ebenso werden Aktualisierungen oder das Neuladen von Inhalten nur nach manueller Betätigung durchgeführt. Der Nutzer wird die App eigenständig kontrollieren und diese Funktionen nach Wunsch ausführen können.



### 4.2. Installation und Einstieg

Für die Entwicklung von einer iPhone App mit Titanium wird ein Intel basierter Computer mit Mac OS X (10.6) benötigt. Existiert die Aufgabe, das Programm zusätzlich auf dem Gerät zu testen, ist selbstverständlich das iPhone und eine Entwickler-Lizenz von Apple Voraussetzung. Diese Lizenz kann für 99 \$ im Jahr beantragt werden. Zudem ist die Lizenz unabdingbar, wenn die Applikation im Apple Store später vertrieben werden soll. Besteht die Anforderung, eine iPhone Applikation lediglich in einem Simulator zu testen und sie nicht zu vertreiben, kann dies kostenlos geschehen. Mit einem gültigen Apple Kundenkonto (die kostenpflichtige Lizenz ist nicht Voraussetzung) werden Software Development Kit (SDK) und Xcode von der Apple Developer Website<sup>14</sup> bezogen und nachfolgend installiert. Danach ist das Titanium Mobile SDK von Appcelerator zu installieren [Titanium 2011]. In Abbildung 4.2 wird das Programm von Titanium nach Auswahl der "Dashboard-Ansicht" gezeigt, welche wichtige Links signifiziert. Diese Links verweisen auf Infor-

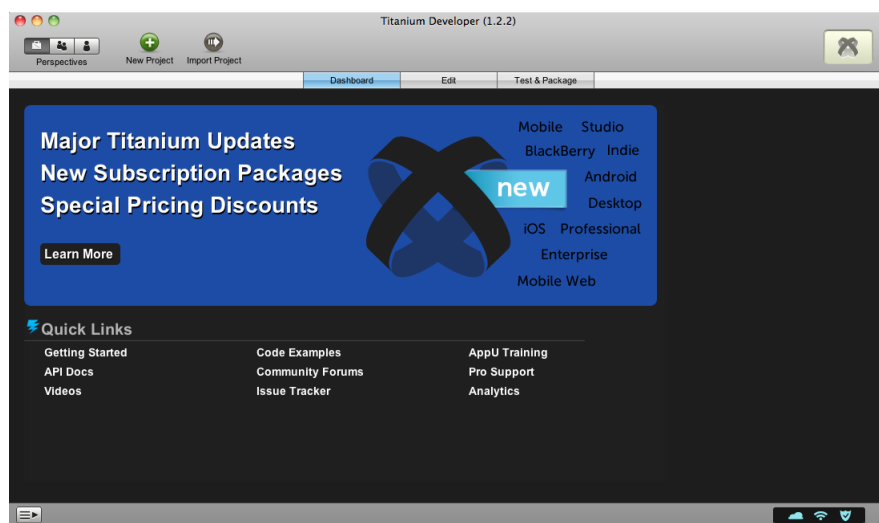


Abbildung 4.2.: Titanium Developer Software (15. August 2011)

mationen und Programmbeispiele, die einen detaillierten und schnellen Einstieg in das Framework ermöglichen. Zur Erleichterung der mobilen Entwicklung empfiehlt

---

<sup>14</sup>Link zu der Apple Developer Website: <http://developer.apple.com/> (Zugriff: August 2011)

## 4.2. Installation und Einstieg

---

es sich, zum einen die Dokumentation der API auf der Website<sup>15</sup> des Anbieters zu verwenden und zum anderen das "Kitchen Sink" Beispielprojekt<sup>16</sup> in das Titanium Programm zu importieren und die Beispiele umfangreich zu studieren, denn in diesem Projekt wird der Leistungsumfang des Frameworks gezeigt.

### 4.2.1. Entwicklungsumgebung

Das Titanium Mobile SDK ist keine Entwicklungsumgebung. In dieser Masterarbeit wird zusätzlich Eclipse<sup>17</sup> als Entwicklungsumgebung verwendet. In Eclipse wird die Implementierung durchgeführt und das Projekt kann mit Titanium Developer direkt auf dem iPhone (wenn die Lizenz vorhanden ist) oder in einem iPhone-Simulator ausgeführt werden.

---

<sup>15</sup>Website der Dokumentation von Titanium: <http://developer.appcelerator.com/apidoc/mobile> (Zugriff: August 2011)

<sup>16</sup>Link zu dem Beispielprojekt: <http://wiki.appcelerator.org/display/guides/Getting+Started+with+Kitchen+Sink> (Zugriff: August 2011)

<sup>17</sup>Es wird empfohlen eine JavaScript basierte Version unter <http://www.eclipse.org/> zu beziehen.

### 4.3. Umsetzung des Konzeptes und Implementierung

Wird mit Titanium Mobile SDK ein neues Projekt angelegt und dieses in Eclipse als File System importiert, weist es die in Abbildung 4.3 gezeigte Struktur auf. In dem "build" Ordner werden von dem Titanium SDK für iPhone und Android

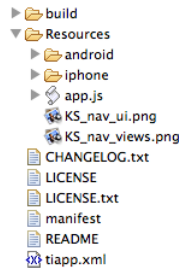


Abbildung 4.3.: Titanium Projekt Struktur

alle nativen Programmelemente separat gespeichert. Für die Programmierung ist "Resources" der relevante Ordner. Hier befinden sich die gerätespezifischen Icons (unterschiedliche Auflösungen) und eine Datei namens "app.js". Diese JavaScript Datei repräsentiert das komplette Applikationsfenster, was im Zusammenhang mit iOS schon in Kapitel 2.2.2 erläutert wurde. Hier müssen alle Sektionen, zum Beispiel einzelne Tabbar-, Navigation- und View-Elemente, angegeben, initialisiert und dem Applikationsfenster hinzugefügt werden.

#### 4.3.1. Tabbar-Elemente

Die Abbildung 4.4 stellt dar, wie die Initialisierung eines Fensters und einer Tabbar programmtechnisch umgesetzt wird. Über die Methode "Titanium.UI.createWindow()" kann ein Fensterobjekt erstellt werden. Dieser Methode sind Attribute anzugeben, wie es der Quellcode exemplifiziert. Beispielsweise wird der gesamte Fensterinhalt samt zugehöriger Programmlogik durch die Angabe von "url" in eine andere JavaScript Datei ausgelagert. Eine "TabGroup" ist die Oberstruktur zum Steuern und Initialisieren verschiedener Fensterinhalte über Tabs, eine Erzeugung erfolgt über die Methode "Titanium.UI.createTabGroup()". Der "TabGroup" werden einzelne "Tab" Objekte hinzugefügt. Ein "Tab" wird über die Methode "Titanium.UI.createTab()" erstellt und erhält durch Attributangaben ein

### 4.3. Umsetzung des Konzeptes und Implementierung

---

```
// create base UI, the root window
var win1 = Titanium.UI.createWindow({
  // title of the Window
  title:'Cityrouter',
  // link to the content
  url:'appWindows/map_window.js',
  // id
  titleid:'map_win_title'
});
// create tab group
var tabGroup = Titanium.UI.createTabGroup();
// create tab element
var tab1 = Titanium.UI.createTab({
  // icon of the tab element
  icon:'img/map.png',
  // name of the tab element
  title:'Map',
  // content provided by the tab
  window:win1,
  // tab id
  titleid:'map_win_title'
});
//add a Tab element with a window UI to the TabGroup
tabGroup.addTab(tab1);
```

**Abbildung 4.4.:** Hauptfenster und Tabbar Initialisierung, Auszug der "app.js"

Fenster-Objekt, Bild und Titel. Diese Vorgehensweise wird für die vier wesentlichen Funktionen und Komponenten, die in Kapitel 4.1.5 definiert wurden, durchgeführt.

#### 4.3.2. Globale Events

Für eine optimierte Struktur und Übersicht des Programmes sind sowohl die Logik, als auch der Aufbau von einzelnen Fenstern in verschiedene Javascript Dateien zu separieren<sup>18</sup>. Zwischen verschiedenen Dateien und somit funktionalen Sektionen besteht nicht die Möglichkeit, auf Objekte oder Dienste der anderen zuzugreifen. Um trotzdem einzelne Sektionen, zum Beispiel bei bestimmten Ereignissen, benachrichtigen zu können, bietet Titanium die Möglichkeit globale Events zu erstellen. In spezifischen Sektionen, die bei Freisetzen des Events benachrichtigt werden sollen, können Listener für das globale Event hinzugefügt werden. Wird das Ereignis an irgendeiner Stelle im gesamten Programmkontext initiiert, werden alle Sektionen mit entsprechendem Listener benachrichtigt. Ein Event ist, wie in Abbildung 4.5 gezeigt, durch den Namen eindeutig gekennzeichnet. Beim Initialisieren des Listeners kann zusätzlich zum Eventnamen eine Funktion angegeben werden.

---

<sup>18</sup>Diese Struktur wird ebenfalls von Appcelerator empfohlen: <http://wiki.appcelerator.org/display/guides/The+Titanium+Architecture> (Zugriff: August 2011)

### 4.3. Umsetzung des Konzeptes und Implementierung

---

<pre>Titanium.App.addEventListener('position', function(e){   //After locating a new place, the position is marked on the map.   drawPositionToMap(e); });</pre>	<pre>Titanium.Geolocation.getCurrentPosition(function(e) {   // fire the Event with new Position-Information   Titanium.App.fireEvent('position', e.coords); });</pre>
(a) Auszug der "openlayersMobile.js"	(b) Auszug der "geolocation.js"

**Abbildung 4.5.:** Event registrieren und ausführen

#### 4.3.3. Systemtypische Buttons und Navigationsbar

Entsprechend der Feststellung in Kapitel 2.2, schätzen Nutzer vor allem die Verwendung von charakterisierenden Systemkomponenten, zudem fördern diese Elemente einen möglichst intuitiven Umgang. Einige Wiedererkennungselemente und iOS typische Elemente wurden in Kapitel 2.2.2 nahegelegt. Wie Titanium den Zugriff auf systemtypische iOS Komponenten ermöglicht, wird anhand der Abbildung 4.6 durch einen Programmabschnitt verdeutlicht. Es wird die Initialisierung von dem "Refresh" und "Spinner" Button gezeigt. Zudem werden diese Buttons für ein Fenster als Navigationsbuttons hinzugefügt. Nach Auswahl des "Refresh" Buttons wird der

```
var refresh = Titanium.UI.createButton({  
  systemButton: Titanium.UI.iPhone.SystemButton.REFRESH  
});  
var spinner = Titanium.UI.createButton({  
  systemButton: Titanium.UI.iPhone.SystemButton.SPINNER  
});  
  
win.setRightNavButton(refresh);  
  
refresh.addEventListener("click", function(e){  
  win.setRightNavButton(spinner);  
  setTimeout(function(){  
    Ti.App.fireEvent('initMap');  
    win.setRightNavButton(refresh);  
  }, 1000);  
});
```

**Abbildung 4.6.:** Navigations Button, Auszug von "map\_window.js"

"Spinner" Button durch einen Timeout eine Sekunde (1000 ms) lang gesetzt. Die Farbe einer Navigationsbar wird über das Attribut "win.barColor" eines Fensters eingestellt.

#### 4.3.4. Kommunikation mit dem Server

Die Kommunikation des Clients mit dem Server erfolgt über SOAP. Titanium stellt einen leichtgewichtigen SOAP-Client bereit, der in eine Sektion einzubinden ist.

### 4.3. Umsetzung des Konzeptes und Implementierung

---

Dies wird durch den Befehl "Titanium.include('suds.js')" durchgeführt. Die Abbildung 4.7 zeigt in einem Programmausschnitt die Initialisierung des SOAP-Clients und den Aufruf einer Methode, die von dem Web Service angeboten wird. Beim Befehl werden der Service-Methode wichtige Werte durch Angabe von Parametern übergeben.

```
// URL to DLR Service
var url = "http://www.cityrouter.de/mobile/services/CityrouterService";
// SOAP-Client init
var suds = new SudsClient({
    endpoint: url,
    targetNamespace: 'http://gateway.cityrouterApp/xsd'
});
// The parameters for a SOAP call to the web service should be handed over
var callparams = {
    userName: Titanium.App.Properties.getString("user"),
    pwd: Titanium.App.Properties.getString("pwd")
};

try {
    suds.invoke('getTrafficObj', callparams, function(xmlDoc){
        var results = xmlDoc.documentElement.getElementsByTagName('getTrafficObjResponse');
```

Abbildung 4.7.: SOAP Kommunikation, Auszug von "monitor.js"

#### 4.3.5. Die Karte als Webview

Durch die CSS-Datei "mapwidgetStyle.css" wird die Darstellung der Karte (eng. map) als HTML-Element formatiert. Lediglich die Karte wird in der HTML als Content angegeben. Für das ganze HTML-Konstrukt und das Map-Objekt wird in den Stylesheets, hinsichtlich der Höhe und Weite, der Wert 100% angegeben. Auf diese Weise wird die maximal mögliche Dimension verwendet und die Webview, je nach Größe, von den Kartendaten dynamisch komplett ausgefüllt.

#### Erzeugung der Map mit Touchfunktionalität

Die Initialisierung des OpenLayers Map-Objektes wird in Abbildung 4.8 gezeigt. Hier ist zu sehen, dass dem Attribut "div" die Identifikation aus der HTML-Datei angegeben werden muss. Mit dem Attribut "sphericalMercator" wird die Verwendung der Mercator Projektion<sup>19</sup> bestimmt [OpenLayers 2011]. Projektionen sind durch eindeutige Codes gekennzeichnet, die von der Euro-

---

<sup>19</sup>Es ist eine Zylinderprojektion, bei der die Erde als eine Sphere behandelt wird.

### 4.3. Umsetzung des Konzeptes und Implementierung

---

```
map = new OpenLayers.Map({
  div: "map",
  sphericalMercator: true,
  maxExtent: new OpenLayers.Bounds(-20037508.34, -20037508.34, 20037508.34, 20037508.34),
  controls: [new OpenLayers.Control.Attribution(), new OpenLayers.Control.TouchNavigation({
    dragPanOptions: {
      interval: 150,
      enableKinetic: true // special effect
    }
  }), new OpenLayers.Control.ZoomPanel()],
  center: new OpenLayers.LonLat(1493661, 6890155), // start position: Berlin
  layers: [dirMapLayer, trafficLayer, textLayer],
  zoom: zoomCenter // start zoomlevel
});
```

**Abbildung 4.8.:** OpenLayers Map mit Touchfunktionalität, Auszug der "openlayersMobile.js"

pean Petroleum Survey Group (EPSG)<sup>20</sup> vergeben werden<sup>21</sup>. Zusätzlich zur Control "OpenLayers.Control.TouchNavigation" wird diesem Objekt die Eigenschaft "enableKinetic" zugeschrieben. "Kinetic" ist der Effekt, durch den beim Ziehen der Karte automatisch etwas weitergeschwungen wird, beziehungsweise die Kartenbewegung ab dem Zielpunkt (Punkt, bis wohin gezogen wurde) abklingt.

#### Karten- und Textmaterial

Hinsichtlich der Beschreibung in Kapitel 4.1.1 gilt es, die separaten Kacheln der Kartenbilder und Textbilder als OSM Objekte in OpenLayers einzubinden. Für diesen Zweck werden in der "OpenStreetMap.js" Datei zwei OSM Objekte erzeugt, sowohl für den Text- als auch für das Kartenmaterial.

Die Initialisierung des OSM Objektes für das Kartenmaterial wird in Abbildung 4.9 gezeigt. Im abgebildeten Programmausschnitt wird ersichtlich, dass beim DLR Server unter "http://www.taxifcd.de/verkehr/tilesNoText/{z}/{x}/{y}.png" alle Textkacheln referenziert sind. Die URL-Elemente, wie {z}, sind Platzhalter und werden nach der Initialisierung anhand der aktuellen Sicht durch einen Algorithmus festgelegt und somit zutreffende Kacheln bestimmt und geladen. Dies geschieht durch die Methode "getURL" im "OpenLayers.Layer.XYZ" Objekt von OpenLayers.

---

<sup>20</sup>Internetpräsenz: <http://www.epsg.org> (Zugriff: September 2011)

<sup>21</sup>Zum Beispiel wird für Breiten- und Längengrad Projektion die Zeichenkette "EPSG:4326" verwendet.

### 4.3. Umsetzung des Konzeptes und Implementierung

---

```
OpenLayers.Layer.OSM.NoText = OpenLayers.Class(OpenLayers.Layer.OSM, {  
    /**  
     * Constructor: OpenLayers.Layer.OSM.NoText  
     *  
     * Parameters:  
     * name - {String}  
     * options - {Object} Hashtable of extra options to tag onto the layer  
     */  
    initialize: function(name, options) {  
        var url = [  
            "http://www.taxifcd.de/verkehr/tilesNoText/{z}/{x}/{y}.png"  
        ];  
        options = OpenLayers.Util.extend({ numZoomLevels: 15, attribution: "" }, options);  
        var newArguments = [name, url, options];  
        OpenLayers.Layer.OSM.prototype.initialize.apply(this, newArguments);  
    },  
    CLASS_NAME: "OpenLayers.Layer.OSM.NoText"  
});
```

Abbildung 4.9.: Zugriff auf Kartenkacheln, Auszug der "OpenStreetMap.js"

#### Verkehrslage

In Kapitel 3.1.1 wurde erläutert, wie die Verkehrslage als MapServer zur Verfügung gestellt wird. Die Initialisierung des MapServers unter Verwendung der FastCGI

```
// create traffic Layer  
trafficLayer = new OpenLayers.Layer.MapServer(trafficName, "http://www.taxifcd.de/fcgi-bin/mapserv.exe", {  
    map: pathToMap + cityNameForPath + "/" + mapName,  
    map_imagetype: 'agga',  
    map_size: [130, 108],  
    imgx: 130 / 2,  
    imgy: 108 / 2  
}, {  
    isBaseLayer: false,  
    singleTile: false,  
    visibility: true,  
    transitionEffect: 'resize'  
});  
if (map.getLayer(trafficLayer) == null) {  
    map.addLayer(trafficLayer);  
}
```

Abbildung 4.10.: Initialisierung der Trafficansicht, Auszug von "openlayersMobile.js"

Ressource wird in dem Programmausschnitt 4.10 abgebildet. Unter dem FastCGI Ordner befindet sich eine Datei mit dem Namen "mapserv.exe", welche die Verkehrslage für OpenLayers als MapServer Objekt zurückliefert. Das Objekt wird der Map als Layer an spezifischer Stelle hinzugefügt, sodass sich der Verkehrslagelayer über dem Kartenmaterial aber unter dem Textmaterial befindet. Dies wird erreicht, indem die Verkehrslage nach dem Kartenmaterial und der Textlayer anschließend hinzugefügt wird.



### 4.3. Umsetzung des Konzeptes und Implementierung

---

#### Abbilden einer Route

Die OpenLayers API stellt mit dem Layer "OpenLayers.Layer.Vector", dem Feature "OpenLayers.Feature.Vector" sowie dem Geometry Objekt "OpenLayers.Geometry.LineString" Komponenten bereit, um eine Route anhand von vielen Punkten auf der Karte darzustellen. Ein Ausschnitt von "openLayersMobile.js" zeigt in Abbildung 4.11, wie die Zeichenkette, die aus einigen durch ein Komma separierten Positionsdaten (in Breiten- und Längengrad) besteht, als eine Route gezeichnet wird. In dieser Sektion wird der Titanium Applikation ein Event-Listener

```
function drawPoints(points){
    //each entry in the list is separated by a comma
    var result = points.split(",");
    var points = [];
    for (var i = 0; i < result.length; i = i + 2) {
        var point = new OpenLayers.Geometry.Point((result[i] * 1), (result[i + 1] * 1));
        // A list obtained from the point, is a Lat/Lon-object of projection EPSG4326.
        // This projection must be converted to the projection EPSG900913.
        // the converted point will be added to an array
        points.push(point.transform(new OpenLayers.Projection("EPSG:4326"), new OpenLayers.Projection("EPSG:900913")))
    }
    //the basis of points can generate a Geometry object
    var geometry = new OpenLayers.Geometry.LineString(points);
    if (lineVector != null) {
        //There is only one route on the map. It removes a previously initialised line.
        routeLayer.removeFeatures([lineVector]);
    }
    lineVector = new OpenLayers.Feature.Vector(geometry, null, {
        strokeColor: "#0033ff",
        strokeOpacity: 0.7,
        strokeWidth: 5
    });
    routeLayer.addFeatures([lineVector]);
}
```

**Abbildung 4.11.:** Route auf Karte abbilden, Auszug von "openlayersMobile.js"

mit der Event-Bezeichnung "draw" hinzugefügt und die in Abbildung 4.11 dargestellte Funktion durchführt. Die Zeichnung einer Route kann in der Anwendung global initiiert werden, indem das Event "draw" ausgelöst und als Wert eine Zeichenkette mit Positionen übergeben wird.

#### 4.3.6. Ortung und FCD-Handhabung

In dem Kapitel 2.5 wurden die einzelnen Ortungsmöglichkeiten genannt. Im Folgenden werden diese Methoden programmtechnisch angewendet. Die Lokalisierung wird auf eine Weise durchgeführt, sodass immer die exakteste Positionsbestimmung verwendet wird.

### 4.3. Umsetzung des Konzeptes und Implementierung

---

#### Digitaler Kompass

Der Boolean-Wert `"Titanium.Geolocation.hasCompass"` gibt Auskunft darüber, ob das Endgerät die Funktionalität eines Kompasses unterstützt. Liegt der Wert als `"true"` vor, kann eine Initialisierung erfolgen und anhand eines Filters (nach Veränderung des Grades) können die Werte des Kompasses kontinuierlich erhalten werden. Damit die Werte nur bei einer Veränderung der Lage um einen bestimmten Grad erhalten werden, ist der Variable `"Titanium.Geolocation.headingFilter"` eine Gradzahl anzugeben (in dieser Arbeit  $90^\circ$ ). Die Methode `"Titanium.Geolocation.getCurrentHeading"` liefert bei Aufruf aktuelle Kompassdaten einmalig zurück. Die anhaltende Datenlieferung erfolgt, wie in

```
var headingCallback = function(e){
  var x = e.heading.x;
  var y = e.heading.y;
  var z = e.heading.z;
  var magneticHeading = e.heading.magneticHeading;
  var accuracy = e.heading.accuracy;
  var trueHeading = e.heading.trueHeading;
  var timestamp = e.heading.timestamp;
};
Titanium.Geolocation.addEventListener('heading', headingCallback);
```

**Abbildung 4.12.:** Kompass Zugriff, Auszug von `"geolocation.js"`

Abbildung 4.12 dargestellt wird, durch Registrierung des `"heading"` Events. Solange das Event registriert ist, werden Kompassdaten basierend auf dem Filter evaluiert.

#### Geopositionierung

Die funktionale Umsetzung der Positionsbestimmung erfolgt ähnlich wie die Kompassbestimmung. Eine Bereitstellung von Lokalisierungsdaten erfolgt einmalig durch die Funktion `"Titanium.Geolocation.getCurrentPosition"` oder kontinuierlich durch die Registrierung eines Events. Für die wiederholende Positionsbestimmung wird dem Geolocation-Modul von Titanium das Event `"location"` registriert. Ebenfalls kann für die kontinuierliche Lokalisierung ein Filter verwendet werden. Es ist ein Distanzfilter mit dem Parameter `"Titanium.Geolocation.distanceFilter"`, durch den in Abhängigkeit von zurückgelegter Strecke in Metern ein Lokalisierungs-Event ausgelöst wird. Nach Ausführung des Events werden die zu verarbeitenden Daten übergeben (siehe Abbildung 4.13). Über die Variable `"Titanium.Geolocation.accuracy"` ist die bevorzugte Genauigkeit anzu-

### 4.3. Umsetzung des Konzeptes und Implementierung

---

```
locationCallback = function(e){
    var longitude = e.coords.longitude;
    var latitude = e.coords.latitude;
    var altitude = e.coords.altitude;
    var heading = e.coords.heading;
    var accuracy = e.coords.accuracy;
    var speed = e.coords.speed;
    var timestamp = e.coords.timestamp;
    var altitudeAccuracy = e.coords.altitudeAccuracy;
    Titanium.App.fireEvent('localisation', e.coords);
    Ti.App.appData = {
        geoData: e.coords
    };
    FCD.addData(e.coords);
};
Titanium.Geolocation.addEventListener('location', locationCallback);
```

**Abbildung 4.13.:** Geolokation Bestimmung, Auszug von "geolocation.js"

geben. Es werden unterschiedliche Angaben bereitgestellt. Hierzu zählen "ACCURACY\_KILOMETER" oder "ACCURACY\_BEST". Für diese Ausarbeitung wird "Titanium.Geolocation.ACCURACY\_BEST" verwendet, damit die genauesten verfügbaren Ortungsinformationen bereitgestellt werden.

#### FCD-Sammlung und -Übertragung

In dem vorherigen Kapitel wurde schon anhand der Abbildung 4.13 deutlich, an welcher Stelle die Lokalisierungsdaten gesammelt und dem FCD-Objekt hinzugefügt werden. Die Sammlung von FCD ist in der Datei "PositionFCD.js" implementiert und diese wird in der Datei "geolocation.js" über "Titanium.include('javascript/PositionFCD.js')" hinzugefügt. Hier werden vor einer Speicherung von Lokalisationsdaten zusätzliche Filter angewendet. Eine Sammlung der Informationen erfolgt, wenn die Daten den Filtereigenschaften entsprechen. Diese programmiertechnische qualitative Verifizierung von FCD wird in Abbildung 4.14 gezeigt.

```
verifyGoodFCD : function(positionData){
    // The precision of the location should satisfy at least be accurate to 13 meters. (GPS)
    if (positionData.accuracy <= 13) {
        // If data elements are already there, it is checked, otherwise add without verification.
        if (this.data.length > 0) {
            // previous item
            var predecessor = this.data[this.data.length - 1];
            // There are a maximum of available data every 10 sec.
            //The risk of redudanten unneeded information is reduced in this way.
            if((new Date(positionData.timestamp).getTime() - new Date(predecessor.timestamp).getTime()) >= 10000){
                return true;
            }
        }
    }
}
```

**Abbildung 4.14.:** Verifizierung von FCD, Auszug aus "PositionFCD.js"

---

## 5. Umsetzung der serverseitigen Komponenten

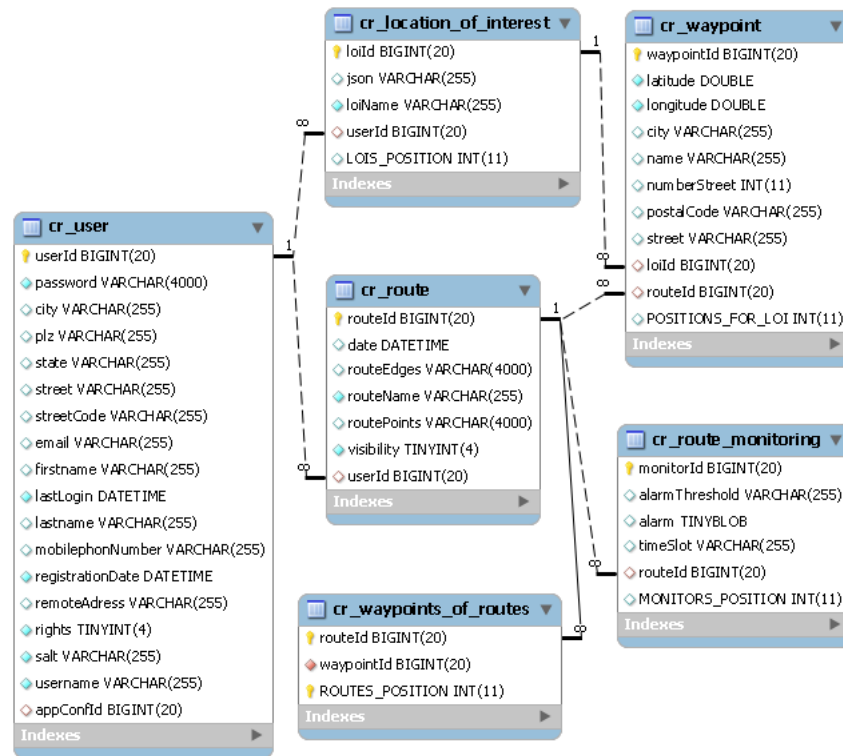
In Hinblick auf Kapitel 4.3 wurde bei dem Konzept und der Implementierung der mobilen Applikation erwähnt, dass die Kommunikation zwischen Client und Server anhand eines einheitlichen und standardisierten Protokolls geschieht. Aufseiten des Servers wird ein Web Service entworfen, der über das Simple Object Access Protocol (SOAP) mit dem Client kommuniziert. Der Web Service wird alle Funktionen bereitstellen, die die mobile Applikation benötigt und dient als Schnittstelle zwischen bereits bestehenden Funktionalitäten des DLR (siehe Kapitel 3.1.1) und eigenen speziellen Applikationsfunktionen für die mobile Applikation. Exemplarisch zählt zu den speziellen Applikationsfunktionen die Methode, bei der kontinuierlich FCD von Smartphonennutzern gesendet und daraufhin von den Web Services verarbeitet wird. Diese Funktionen müssen auf geeignete Weise eingebunden und dem Client zur Verfügung gestellt werden.

### 5.1. Datenmodell

Um personalisierte Routen speichern und bereitstellen zu können, wird ein Datenmodell erstellt. Diesbezüglich ist es notwendig ebenfalls einen Nutzer zu kreieren, für den die personalisierte Route bereitgestellt wird. Mit personalisierten Routen sind für die Routenüberwachung von einem Nutzer gespeicherte Verkehrsstrecken gemeint. Die von einem Anwender gespeicherten Wege werden durch die mobile Applikation angefordert. Daraufhin wird für jede Route durch den Travel-Time-Service serverseitig die aktuelle Verkehrssituation auf der Strecke berechnet. Diese Informationen werden nachfolgend zur mobilen Applikation übertragen.

## 5.1. Datenmodell

### 5.1.1. Beschreibung der Datenbankstrukturen



**Abbildung 5.1.:** Entwurf der Datenbankstrukturen als Entity Relationship Modell

In Abbildung 5.1 wird das Entity Relationship Modell (ERM) mit den Datenbankstrukturen dargestellt. Einem Nutzer Entity "cr\_user" wird bei der Anmeldung eine eindeutige Identifikationsnummer (Primarykey) von der Datenbank zugeordnet. Ebenfalls müssen das Passwort, das Registrierungsdatum, die E-Mail-Adresse, der letzte Anmeldezeitpunkt und Nutzernamen angegeben werden. Bei einer Registrierung ist die letzte Anmeldung identisch zum Anmeldedatum. Die E-Mail-Adresse wird benötigt, um den Anwender in Zukunft über wichtige Informationen und Funktionen zu informieren. Ein Nutzer kann mehrere Entitys "cr\_location\_of\_interest" besitzen. Der Location of Interest (LOI) ist ebenfalls eine Identifikationsnummer als Primarykey gesetzt. Der Name eines LOI ist eine Pflichtangabe. Ein LOI ist die geografische Gegend, die von einer Person definiert und gespeichert wurde. Beispielswei-

## 5.1. Datenmodell

---

se kann ein Haus als LOI durch mehrere Wegpunkte (eng. waypoints) mit Breiten- und Längengrad angegeben werden. In diesem Fall ist der LOI als ein geografischer Umkreis definiert. Ebenfalls kann ein LOI aber auch ein einziger Wegpunkt mit einer Adresse sein (Stadt, Straße, Hausnummer, Postleitzahl). Die Breiten- und Längengrade eines Punktes müssen dabei immer gesetzt sein. Mehrere Waypoints können zu einigen unterschiedlichen Routen zusammengesetzt werden, aber jede Route gehört einem Nutzer. Durch das Element "cr\_route\_monitoring" wird es zukünftig möglich sein, dass Anwender sich zu einem bestimmten Zeitabschnitt über den Verkehrszustand der überwachten Strecken informieren können.

### 5.1.2. Umsetzung

Die Umsetzung der Datenstrukturen erfolgt mit Hibernate<sup>1</sup>. Es ist ein Open-Source Framework basierend auf Java. Die Stärke von Hibernate ist das Object-Relational Mapping (ORM). Hibernate ermöglicht es, die Entitys als Plain Old Java Object (POJO)<sup>2</sup> zur Verfügung zu stellen. Ebenfalls werden für die Handhabung und Abfragen der Datenbank keine Data Manipulation Language (DML)<sup>3</sup> oder Data Definition Language (DDL)<sup>4</sup> direkt<sup>5</sup> benötigt, stattdessen kann einfach der Zugriff mittels Java-Methoden der Objekte erfolgen. Dies zeigt sich im Vergleich zu Java Database Connectivity (JDBC) als Vorteil. Unter Verwendung von JDBC ist es lediglich möglich, eine Datenbankverbindung aufzubauen, SQL-Anfragen an die Datenbank weiterzuleiten und Ergebnisse in Java bereitzustellen. Ein weiterer signifikanter Vorteil von Hibernate ist die Abstraktion der verwendeten Datenbank und Datenstrukturen. Einstellungen von Hibernate werden in einer XML Datei namens "hibernate.cfg.xml" gespeichert. Hier werden beispielsweise die Nutzerdaten (Benutzername und Passwort) für die Datenbank, die Sprache der Datenbank sowie die zu erzeugenden Elemente (mit Verweis auf die Java-Klassen) definiert<sup>6</sup>. Durch das Attribut "hibernate.dialect" ist der Dialekt einzustellen. Mit den Attri-

---

<sup>1</sup>Link zum Framework: <http://www.hibernate.org> (Zugriff: September 2011)

<sup>2</sup>Es ist ein normales Java-Objekt, dem keine speziellen Konventionen vorliegen.

<sup>3</sup>Sprachkonstrukte für die Behandlung und den Umgang von Daten, die in Strukturen der Datenbank vorliegen.

<sup>4</sup>Sprachkonstrukte für die Behandlung und den Umgang von Datenbankstrukturen.

<sup>5</sup>Hibernate übernimmt die Aufgabe, mit DML und DDL den Zugriff zu steuern.

<sup>6</sup>Verfügbare Einstellungen werden in der Dokumentation von Hibernate genannt: <http://docs.jboss.org/hibernate/core/3.3/reference/en/html/session-configuration.html#configuration-optional> (Zugriff: September 2011)

## 5.1. Datenmodell

---

buten "connection.driver\_class" und "connection.url" wird die Verbindung zu der Datenbank angegeben. Somit besteht die Möglichkeit, flexibel und automatisch ein anderes Datenbanksystem zu verwenden. Diese Abstraktion ermöglicht eine bedeutende Dynamik für die Entwicklung von Datenbankstrukturen. Das in dieser Arbeit umgesetzte Datenmodell ist mittels Hibernate demzufolge für unterschiedliche Datenbanksysteme<sup>7</sup> agil einsatzbereit.

### Implementierung

Anhand von Java-Klassen und Annotationen lassen sich unter Verwendung von Hibernate die Elemente, deren Attribute und Beziehungen als Datenbankstrukturen definieren. Annotationen werden mit einem "@" Zeichen in den Java-Quelltext eingebunden und bilden Metainformationen. Im Zusammenhang mit Hibernate werden als Annotationen die Java-Elemente der Java Persistence API ("javax.persistence")<sup>8</sup> standardmäßig verwendet. Alternativ können die Metainformationen der Elementeigenschaften auch per XML verfasst werden. Annotationen werden in der Java-Datei direkt an der Stelle verzeichnet, wo sie die entsprechenden Eigenschaften von Parametern und Beziehungen definieren. Aufgrund dessen unterstützen Annotationen die Übersichtlichkeit durch strukturierte Angaben. In Abbildung 5.2 wird ein Programmausschnitt des Datenbankelementes einer Applikationsroute gezeigt.

```
@Entity
@Table(name="cr_route")
public class CRRoute implements IDatabasePOJO {
    /*
     * column definition start
     */
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "routeId", updatable = false, unique = true, nullable = false)
    private Long routeId;
    @Column(name = "routeName", unique = false, nullable = false)
    private String routeName;
    @ManyToOne
    @JoinColumn(name = "userId", updatable = false, nullable = true)
    private CRUser user;
```

**Abbildung 5.2.:** Implementierung eines Datenstruktur-Elements anhand von Hibernate und Annotationen

---

<sup>7</sup>Beispielsweise MySQL, Oracle oder PostgreSQL

<sup>8</sup>Link zu der Dokumentation: <http://java.sun.com/javase/5/docs/api/javax/persistence/package-summary.html> (Zugriff: September 2011)

## 5.1. Datenmodell

---

Anhand des Annotation-Types "Table" wird der Tabellename festgelegt. Durch Angabe der Annotation "Id" wird die Identifikationsnummer der Route als Primarykey definiert. Mittels "GeneratedValue" kann festgelegt werden, wie die Identifikation erzeugt wird und durch die Angabe des Wertes "GenerationType.AUTO" erfolgt die Erstellung in Abhängigkeit von der Strategie einer Datenbank<sup>9</sup> automatisch. Der Annotation-Type "Column" definiert die Spaltenwerte einer Tabelle. Es kann der Name einer Spalte und deren Eigenschaften, wie "modifizierbar" oder "einzigartig", gesetzt werden. Die Beziehungen zwischen den Elementen werden bei Hibernate durch die Annotationen "ManyToMany", "OneToMany", "ManyToOne" oder "OneToOne" angegeben. Die Erstellung der Datenbanktabelle geschieht durch eine Bestimmung der Java-Klasse als Mapping in der Hibernate Konfigurationsdatei (siehe Abbildung 5.3). Das Hibernate Tool "hbm2ddl" führt von den angegebenen Mapping Klassen eine Umsetzung zu der Data Definition Language (DDL) der festgelegten Datenbank durch. Beim Tool wird "update" als Inhalt angegeben, alternativ ist auch "create"<sup>10</sup>, "create-drop"<sup>11</sup> oder "validate"<sup>12</sup> möglich. Durch "update" werden Datenbanktabellen und der Dateninformationsgehalt beibehalten, indem lediglich eine Modifizierung überprüft und bei vorhandenen Änderungen durchgeführt wird.

```
<!-- e.g. validate | update | create | create-drop -->
<!-- we will use 'update' to store tables of the application and just update informations-->
<property name="hibernate.hbm2ddl.auto">update</property>
<mapping class="de.dlr.ts.cityrouter.data.persistence.CRUser" />
```

**Abbildung 5.3.:** Hibernate Konfiguration: Erstellen von Datenbankstrukturen anhand von Java-Klassen, Ausschnitt der Datei "hibernate.cfg.xml"

---

<sup>9</sup>Beispielsweise bei MySQL ist die Default-Strategie "auto\_increment". Die Identifikationszahl wird automatisch hochgezählt.

<sup>10</sup>Tabellen werden bei Start der Serverapplikation neu erstellt. Falls in dem verwendeten Datenbankschema bereits gleichnamige Tabellen vorhanden sind, werden diese entfernt.

<sup>11</sup>Zusätzlich zum Ablauf bei "create" wird das Schema nach der Beendigung einer Session entfernt.

<sup>12</sup>Das Schema wird validiert und keine Änderungen werden durchgeführt.

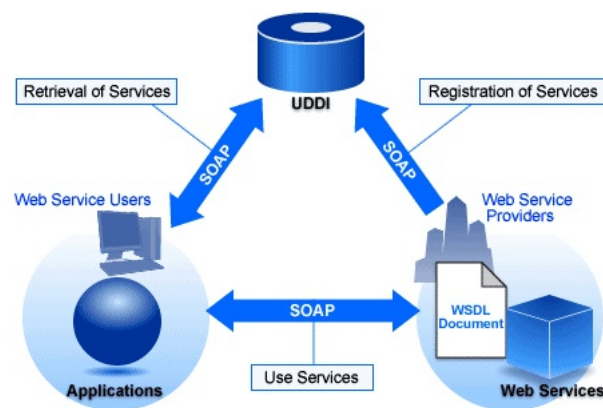


### 5.2. Web Service

Ein Web Service ist eine funktionale Schnittstelle, durch die (programmtechnische) Dienste im Web bereitgestellt werden. Laut W3C kennzeichnet einen Web Service weiterhin Folgendes:

*"Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. Web services are characterized by their great interoperability and extensibility, as well as their machine-processable descriptions thanks to the use of XML."* [W3C 2002]

Somit unterstützen Web Services laut W3C die Erweiterbarkeit der Dienste und die Interoperabilität zwischen verschiedenen Anwendungsprogrammen, die auf unterschiedlichen Plattformen und/oder Frameworks aktiv sein können. Nach der Definition des W3C besitzt ein Web Service diese Eigenschaft zugunsten der Verwendung von Extensible Markup Language (XML)<sup>13</sup>. Ein Web Service besteht aus den drei in Abbildung 5.4 dargestellten Bestandteilen. In einem Universal Description, Dis-



**Abbildung 5.4.:** Web Service - die 3 elementaren Bestandteile

covery and Integration (UDDI) Register werden die Dienste von Anbietern veröffentlicht. Ein Service wird durch die Web Services Description Language (WSDL)<sup>14</sup>

<sup>13</sup>Spezifikation des WC3: <http://www.w3.org/TR/xml/> (Zugriff: August 2011)

<sup>14</sup>WSDL ist vom W3C standardisiert. Link zum Standard: <http://www.w3.org/TR/wsdl> (Zugriff: September 2011)

## 5.2. Web Service

---

definiert. Die im Register veröffentlichten Dienste sind für Nutzer auffindbar und können gebunden und verwendet werden. Die Kommunikation erfolgt mit dem Protokoll SOAP. Wie bereits in Kapitel 2.4.1 erwähnt, basiert die Traffic-Data-Plattform des DLR auf einer SOA. Die Web Services beruhen auf dem Systemparadigma einer SOA und werden in der TDP eingesetzt. Deshalb wird im Folgenden ebenfalls ein Web Service kreiert.

### 5.2.1. Web Service Funktionalitäten für die App

Grundsätzlich werden bei dieser Arbeit mittels des Web Services drei Funktionen angeboten:

- Routenüberwachung - Client kann überwachte Routen anfragen, Server erfragt die Daten beim Travel-Time-Service.
- FCD-Sendung - jeder Nutzer sendet kontinuierlich FCD an den Server. FCD wird als JSON gesendet (siehe Kapitel 4.1.1 und Kapitel 4.3.6). Die Sendung wird geeignet verarbeitet und in einer Datenbank des DLR gespeichert.
- Routing - Client kann beim Server eine verkehrsoptimierte Route von einem Start- zum Zielpunkt und optional mit einem Zwischenpunkt anfragen. Der Web Service leitet diese Anfrage an die Routing-Engine weiter. Daraufhin antwortet die Routing-Engine mit einer berechneten Route als Zeichenkette, die an den Nutzer übermittelt wird.

Der Web Service wird anhand eines Ant-Skripts<sup>15</sup> automatisch erzeugt. Als Ant-Skript wird die "build.xml" Datei bezeichnet, die von Ant ausgeführt wird, in der wichtige Einstellungen und Tasks zu definieren sind [Schmidt 17. Juli 2009, S. 55]. Ein Web Service kann mittels Ant automatisch anhand einer vorliegenden Java-Klasse erstellt werden, indem zusätzlich Apache Axis2<sup>16</sup> verwendet wird. Dieses Framework bietet unter Verwendung von "java2Wsd" eine Funktionalität an, die die Web Service Beschreibung adäquat zu einer Java-Klasse erzeugt. "java2Wsd" erstellt somit die WSDL, die auf XML basiert. In der WSDL werden Funktionen der Java-Klasse als Methoden des Web Services definiert. Des Weiteren werden

---

<sup>15</sup>Internetpräsenz von Apache Ant: <http://ant.apache.org/> (Zugriff: September 2011)

<sup>16</sup>Website des Frameworks: <http://axis.apache.org/axis2/java/core/> (Zugriff: September 2011)

## 5.2. Web Service

---

Variablen, die der Java-Funktion übergeben werden müssen, als gleichnamige Parameter der Web Service Methode festgelegt. Zudem wird durch Axis2 und dem Ant-Skript für den Service ein Axis Archive (AAR) automatisch erstellt. Die AAR-Datei beinhaltet die kompilierten Programme und die Beschreibung des Dienstes. Der Web Server Apache Tomcat fungiert beim DLR als Container für die Services [Schmidt 17. Juli 2009, S. 56]. Das AAR muss dem Web Server hinzugefügt werden. Daraufhin ist der Service registriert und steht für Zugriffe (Client-Server Kommunikation) zur Verfügung.

Im Web existieren einige Open-Source Projekte für die Handhabung von JSON mittels Java<sup>17</sup>. Für diese Arbeit wird GSON<sup>18</sup> verwendet. Dieses Framework ermöglicht es, ein Objekt zu einem JSON String durch einen simplen Methodenaufruf "toJson()" umzuformen. Durch die Methode "fromJson()" wird anhand eines vorliegenden JSON Strings ein entsprechendes Objekt erzeugt. Mit dieser Methode kann ebenfalls eine Gruppierung von Objekten aus einem JSON-String erstellt werden.

### FCD Verarbeitung

Die mobile Applikation sendet Positionsdaten als 2-dimensionalen JSON-Array. Eine Übertragung kann auf diese Weise mehrere FCD-Objekte enthalten. Sind die Daten fehlerhaft oder traten bei der Speicherung Fehler auf, wird der Client darüber benachrichtigt, dass die Sendung nicht erfolgreich verarbeitet wurde. Es wird aufseiten des Servers eine Java-Klasse erstellt, deren Parameter identische Bezeichnung und Variablen-Typen hat wie ein JavaScript FCD-Objekt. Folgend wird ein Array von Objekten dieser Java-Klasse durch einfache Umformung des JSON-Arrays gewonnen. Dieser Funktionsablauf wird in Abbildung 5.5 dargestellt. Die Traffic Data

```
public String sendFCDData(String data) {  
    FCDData[] fcdDatas = new Gson().fromJson(data, FCDData[].class);  
    for (FCDData fcdData : fcdDatas) {  
        //...
```

**Abbildung 5.5.:** Web Service: JSON-String zu FCD Objekten formen

Plattform (TDP) des DLR wurde in Kapitel 2.4.1 erläutert. Diese Plattform verknüpft Prozesse und Dienste im Zusammenhang mit Verkehrsdaten und ermöglicht

---

<sup>17</sup>Beispielsweise JSON-Lib <https://github.com/aalmiray/Json-lib> (Zugriff: September 2011)

<sup>18</sup>Link zu dem Framework: <http://code.google.com/p/google-gson/> (Zugriff: September 2011)

## 5.2. Web Service

---

die Interoperabilität der verschiedenen Komponenten. Als Datenbank wird bei der TDP PostgreSQL<sup>19</sup> eingesetzt. Unter der Verwendung des Java Database Connectivity (JDBC) für PostgreSQL<sup>20</sup> kann der Web Service eine Verbindung zu der Datenbank herstellen. Datenbank-Operationen<sup>21</sup> werden durch JDBC mittels Java ausgeführt. Die Werte von jedem FCD-Objekt werden in der PostgreSQL Datenbank mit der "INSERT"<sup>22</sup> Datenbankoperation eingefügt.

### 5.2.2. Routing-Engine und Routen-Monitoring

In Kapitel 3.1.1 wurde erläutert, dass Routing und Routen-Monitoring<sup>23</sup> beim DLR durch die Anwendung von Apache ActiveMQ mittels eines Message Broker bereitgestellt werden. Eine Applikation kann sich im internen Netzwerk des Instituts, also im verteilten System, beim Message Broker als Client für einen Dienst anmelden. Es wird ein Java Archive namens "FCDMessaging.jar" intern zur Verfügung gestellt, das benötigte Klassen und Komponenten für einen Kommunikationsaufbau bereitstellt.

#### Umsetzung Routing

Im Programmausschnitt 5.6 wird gezeigt, wie die Java-Klassen "FCDMessageRequester" aus der "FCDMessaging.jar" zu initialisieren sind, um eine Verbindung mit den Diensten herzustellen. Die Identifikation des Routing Service ist "FCD.RoutingService.berlin" und für den Travel-Time-Service wird "FCD.TTS@berlin" verwendet. Mit diesen eindeutigen Identifikationen (beim DLR eine Zeichenkette) melden sich die Services bei dem Sendesystem (Message Broker) an, durch Angabe des Namens ist dem Message Broker bekannt, zu welchem Dienst eine Kommunikation für den Client hergestellt wird. Des Weiteren wird in der Abbildung 5.6 mit der Methode "getRoute()" aufgezeigt, wie den Smartphone-Nutzern die Funktionalität nun über den Web Service zur Verfügung gestellt wird und die Kommunikation abläuft. Ein Client ruft über SOAP diese Methode auf und über-

---

<sup>19</sup>Website des Anbieters: <http://postgresql.de/> (Zugriff: September 2011)

<sup>20</sup>Link zum PostgreSQL JDBC Treiber: <http://jdbc.postgresql.org/> (Zugriff: September 2011)

<sup>21</sup>z.B. "SELECT \* FROM tabellen\_name;" selektiert den kompletten Inhalt einer Datenbanktabelle mit dem Namen "tabellen\_name"

<sup>22</sup>Link zu der PostgreSQL Dokumentation: <http://www.postgresql.org/docs/8.1/static/sql-insert.html> (Zugriff: September 2011)

<sup>23</sup>Wie in Kapitel 3.1.1 genannt, wird Routen-Monitoring intern als "Travel-Time-Service" bezeichnet.

## 5.2. Web Service

---

```
public CityrouterService() {
    // Configuration properties
    ConfigurationProperties.init("resources/META-INF/configuration.properties");
    postgreeHandler = new DBPostgreeHandler();
    // Message Requester for the Routing-Service
    messageRequesterRouting = new FCDMessageRequester(
        "FCD.RoutingService.berlin");
    // Message Requester for Travel-Time-Service
    messageRequesterTTS = new FCDMessageRequester("FCD.TTS@berlin");
}
/**
 * Method to calculate a route
 * @param route Route the query string must be passed
 * @return The calculated route as a string
 */
public String getRoute(String route) {
    Object obj = requestMessagingSystem(FCDMessage.GET_ROUTE,
        messageRequesterRouting, route);
    return obj.toString();
}
```

**Abbildung 5.6.:** Web Service: Initialisierung der Message Service und Routing Zugriff, Ausschnitt der Datei "CityrouterService.java"

gibt eine Routenanfrage als Zeichenkette. Daraufhin leitet der Web Service diese Anfrage an den Message Broker weiter. Der Routing-Service berechnet die Route für die Anfrage und gibt sie als Zeichenkette zurück. Der Web Service leitet diese Zeichenkette an den Client weiter.

### Umsetzung Routen-Monitoring

In der Abbildung 5.7 ist die Methode "getTrafficObj()" dargestellt. Diese Methode wird von der Smartphone Applikation per SOAP aufgerufen, wenn die Routenüberwachung angezeigt oder aktualisiert wird. Um eine Überwachung von personalisierten Routen zu ermöglichen, muss dieser Methode der Nutzernamen und das Passwort übergeben werden. Anhand des Passwortes wird sichergestellt, dass die personalisierten Routen nur von entsprechenden validierten Nutzern eingesehen werden. Nach Aufruf der Methode werden der Nutzernamen und das Passwort verifiziert. Die gespeicherten Routen des Nutzers werden aus der Datenbank gelesen, wenn die Verifizierung eine Gültigkeit feststellt. Jeder Nutzer kann beliebig viele Routen besitzen. Die aktuelle Verkehrssituation auf einer Route wird durch den Travel-Time-Service bestimmt, indem der Dienst eine Zeichenkette von Kanten erhält. Für diese Zeichenkette ist ein Format definiert, bei dem jede Kante einer Route durch ein Komma getrennt ist. Weiterhin fordert das Format, dass die Kanten von verschiedenen Routen durch einen senkrechten Strich ("|") getrennt werden. Der Service berechnet die

## 5.2. Web Service

---

```
public String getTrafficObj(String userName, String pwd) {  
    if (validateUser(userName, pwd)) {  
        String returnStr = "[";  
        IDAO userDao = new CRUserDAO();  
        CRUser user = (CRUser) ((CRUserDAO) userDao).getByName(session,  
            userName);  
        IDAO routeDao = new CRRouteDAO();  
        List<CRRoute> routes = ((CRRouteDAO) routeDao).getRoutesForUser(  
            session, user);  
        String edges = "";  
        for (int i = 0; i < routes.size(); i++) {  
            CRRoute route = routes.get(i);  
            if (!edges.equals("")) {  
                edges += "|";  
            }  
            edges += route.getRouteEdges().replaceAll(" ", ",");  
        }  
        Object obj = requestMessagingSystem(FCDMessage.GET_TRAVELTIMES,  
            messageRequesterTTS, edges);  
    }  
}
```

**Abbildung 5.7.:** Web Service: Routen Überwachung, Ausschnitt der Datei  
"CityrouterService.java"

aktuelle Reisezeit, die Streckenlänge und mögliche Verzögerungen für jede Route. Mit Verzögerungen sind Positionen auf der Strecke gemeint, die der Service identifiziert und kennzeichnet. An diesen spezifischen Stellen treten durch Stau oder hohes Verkehrsaufkommen Zeitverzögerungen auf. Diese Informationen werden der mobilen Applikation unter Verwendung des kompakten Datenformats JSON übertragen.

### 5.2.3. OpenStreet Map Offline-Karten

Der Istzustand des DLR Kartenmaterials wurde in Kapitel 3.1.1 beschrieben. Die Text- und Kartenbilder werden separat kreiert und stehen durch einen Server des DLR für den Internetzugriff zur Verfügung (siehe Kapitel 4.3.5). Wie bereits in Kapitel 4.1.1 erläutert, bietet das OpenLayers Objekt "OpenLayers.Layer.OSM" die Möglichkeit, Kacheln in einem gitterähnlichen Konstrukt zu laden. Jede Kacheldatei ist im Gitter durch x, y (Zeile, und Spalte) sowie z (Zoomstufe) eindeutig identifizierbar. Anhand des betrachteten Ausschnittes in der Karte und der gewählten Zoomstufe werden benötigte Kacheln berechnet und über das Internet bezogen. Sind die Karteninformationen also nicht lokal gespeichert, müssen sie zuerst während der Nutzung der Smartphoneapplikation übertragen werden. Hierdurch wird ein hohes Datenvolumen verursacht, da jedes benötigte Kartenbild zu übertragen ist. Bei einer Veränderung der Zoomstufe oder des Kartenausschnitts werden zutreffende Kacheln nachgeladen.

## 5.2. Web Service

---

Die Speicherhierarchie der Kacheldaten sieht dabei folgendermaßen aus: Die Zoomstufe wird als Ordner erstellt und die x-Werte als Unterordner der jeweiligen Zoomstufe. Die z-Werte werden als Name der Bilddatei verwendet. Das DLR erstellt die Kartendaten für Deutschland und an Deutschland grenzende Staaten mit 16 Zoomstufen (Zoom 0 - 15). Es ergeben sich 16 Ordner für die Zoomstufen, in diesen Ordnern befinden sich zusammengerechnet 11250 Unterordner, die durch die x-Werte gebildet werden. Insgesamt sind es 16695955 Bilddateien, die ein Speichervolumen von ungefähr 65,9 Gigabyte (GB) umfassen. Hieraus wird deutlich, dass die Karten- und Textdateien einen äußerst großen Datenbestand bilden. Es ist nicht möglich, Kartenbilder in dieser Größe bei einem Smartphone lokal zu speichern, da zum einen die Speicherkapazität bei vielen Geräten nicht ausreicht und zum anderen eine derartig hohe Speicherbelegung unpraktikabel und den Nutzern nicht zumutbar ist. Wie können nun jedoch Kartendaten speziell für Berlin von den DLR-Kacheln erhalten und somit die benötigte Speicherkapazität deutlich reduziert werden? Zur Lösung dieser Aufgabestellung wird ein Java Programmcode geschrieben, der die Bilddateien des DLR für eine bestimmte Eingrenzung lädt und lokal abspeichert. Die lokal gespeicherten Dateien werden in das Smartphone Projekt manuell eingebunden. Ein Ausschnitt dieser Implementierung wird in Abbildung 5.8 gezeigt. Um den Datenbestand von Kartenkacheln möglichst gering zu halten, ist es von signifikanter Bedeutung, einen begrenzten Ausschnitt zu wählen. Die Werte für die Kartengrenze werden bestimmt, indem die Titanium Logging-Funktion `"Titanium.API.info('value')"`<sup>24</sup> zur Hilfe genommen wird. Durch eine Applikationsausführung mit dem Simulator werden vom DLR bezogene Bilddaten zugunsten dieser Funktion protokolliert. Zur Begrenzung des Kartenausschnittes bietet es sich an, die südwestliche und nordöstliche Grenze durch zwei Städte festzulegen. Bei dieser Entwicklung wird dafür Brück (im Südwesten von Berlin) und Bernau (im Nordosten von Berlin) festgelegt. Zudem wird die Datenkapazität weiterhin reduziert, indem die Anzahl der möglichen Zoomlevel minimiert wird. Beispielsweise kann mit Zoomlevel 0 die ganze Welt betrachtet werden, sodass eine Verwendung ab Zoomlevel 5 sinnvoll erscheint, um lediglich als minimalen Zoom einen Überblick über Deutschland zu ermöglichen. Folglich werden bei diesen Städten anhand der Protokollierung die Werte für 11 Zoomstufen (Zoomstufe 5 - 15) identifiziert.

---

<sup>24</sup>*value* ist hier als Beispiel angegeben. In der Konsole wird *value* ausgegeben.

## 5.2. Web Service

---

```
// leftX and maxY (Brück near Berlin) *** rightX and minY (Bernau near Berlin)
private final static int[][] boundValues = { { 16, 9, 20, 15 },
{ 32, 20, 36, 23 }, { 67, 38, 71, 44 },
{ 136, 82, 140, 85 }, { 270, 164, 280, 174 },
{ 546, 332, 553, 339 }, { 1096, 668, 1103, 677 },
{ 2196, 1339, 2205, 1348 }, { 4390, 2683, 4408, 2695 },
{ 8790, 5367, 8811, 5387 }, { 17595, 10742, 17625, 10776 } };

public static void main(String[] args) {
    xBoundLeft = boundValues[0][0];
    xBoundRight = boundValues[0][2];
    yBoundHeightMin = boundValues[0][1];
    yBoundHeightMax = boundValues[0][3];
    for (int i = 0; i < zoomLevels.length; i++) {
        xBoundLeft = boundValues[i][0];
        xBoundRight = boundValues[i][2];
        yBoundHeightMin = boundValues[i][1];
        yBoundHeightMax = boundValues[i][3];

        for (int startX = xBoundLeft; startX <= xBoundRight; startX++) {
            for (int startY = yBoundHeightMin; startY <= yBoundHeightMax; startY++) {
                int zoom = zoomLevels[i];
                String fileName = startY + ".png";
                String additionalPath = zoom + "/" + startX + "/";
                createImagesForNotTextTitles(zoom, startX, startY,
                    additionalPath, fileName);
                if (zoom > 5) // DLR create the text information only for zoom levels above 5
                    createImagesForTextTitles(zoom, startX, startY,
                        additionalPath, fileName);
            }
        }
    }
}
```

**Abbildung 5.8.:** OSM Karten offline speichern - Ausschnitt von "OpenLayersImageLoader.java"

Die Stadt Brück im Südwesten von Berlin liefert für jede Zoomstufe die maximale y-Koordinate sowie die minimale x-Koordinate. Hingegen ergibt die Stadt Bernau im Nordosten von Berlin die minimale y-Koordinate und maximale x-Koordinate für die jeweilige Zoomstufe. Das Java Programm erzeugt für diesen Kartenausschnitt mit 11 Zoomstufen 266 Ordner und 4405 Bilddateien, deren Speicherbelegung 28.1 Megabyte (MB) beträgt.



---

## 6. Testphase und Identifizierung von Qualitätsmerkmalen

Im Institut für Verkehrssystemtechnik des DLR e.V. wurden iPhones mit der iOS Version 4 zwei Mitarbeitern anvertraut und die Applikation auf diesen Geräten installiert. Es besteht die Anforderung, die Anwendung lediglich durch diese Mitarbeiter intern im Institut zu überprüfen und auf den vorhandenen Geräten zu installieren. Wie im Ausblick 7.3 noch genannt wird, wurde durch den betrieblichen Betreuer bei der Standortleitung ein notwendiger Antrag für eine verbreitete Verteilung der Applikation im DLR gestellt. Der Genehmigungsprozess wurde von dem Betreuer als langjährig bezeichnet und war zum Ende dieser Ausarbeitung nicht abgeschlossen. Die Mitarbeiter wurden beauftragt, die Software im alltäglichen Gebrauch zu nutzen. Dies wurde bereits ab Anfang August mit ersten Software Ergebnissen durchgeführt, um ein frühzeitiges Feedback zu erhalten und auf Verbesserungsvorschläge eingehen zu können. Die hieraus gewonnenen Resultate und Anregungen werden im Folgenden nahegelegt. Weiterhin wird die Erreichung der gestellten Arbeitsziele in diesem Kapitel validiert. Für diese genannten Punkte wird mittels der Testphase eine Überprüfung durchgeführt:

- FCD wird erfasst und an die Anwendung des DLR gesendet.
- Die Applikation bietet durch eine leichte Bedienbarkeit und Bereitstellung interessanter, aktueller Verkehrs- und Routeninformationen einen vernehmbaren Mehrwert.
- Die Entwicklung erfolgt mit einem möglichst plattformübergreifenden Framework, um eine Portierung für Android zu erleichtern.

### 6.1. Überprüfung von der FCD-Erhebung

Das Ziel dieser durchgeführten Tests ist ein Nachweis der technischen Lauffähigkeit und der korrekten gelieferten Ergebnisse in Bezug auf die FCD-Erfassung und -Übertragung zum DLR Server. In Kapitel 5.2.1 wurde erläutert, wie die periodisch erhobenen FCD von Smartphones serverseitig in einer Datenbank des DLR gespeichert werden. Um die korrekte Sammlung und Speicherung der Daten zu validieren, bietet es sich an, die gespeicherten Informationen der Datenbank visualisiert aufzuzeigen. Diese Visualisierung erfolgt mithilfe der Keyhole Markup Language (KML). Die Sprache basiert auf XML und liegt in der Version 2.2<sup>1</sup> als Standard des Open Geospatial Consortium (OGC)<sup>2</sup> vor. KML ermöglicht es, Geodaten zu beschreiben und grafisch in Google Earth<sup>3</sup> darzustellen. Bei einem Microsoft Disk Operating System (MS-DOS) wird unter Verwendung eines Microsoft Batch Skriptes<sup>4</sup> automatisiert die Erstellung einer KML-Datei, für in der Datenbank hinterlegten Geodaten, durchgeführt. Das Microsoft Batch Skript wird in Anhang D.2 gezeigt. Nach Ausführung des abgebildeten Skripts sind über die Kommandozeile die Eingabe der Identifikationsnummer eines Fahrzeuges und das Datum, an dem die Strecke gefahren wurde, erforderlich. Anhand der Eingabedaten wird eine KML-Datei erzeugt. Durch die festgestellte Geschwindigkeit bei einer Geoposition wird die Farbe bestimmt, die für eine grafische Darstellung der Geodaten verwendet wird. Die Geschwindigkeit wird gerundet, indem sie mit fünf addiert, anschließend die Summe durch 10 dividiert und der Wert des Quotienten nachfolgend mit 10 multipliziert wird. Es wird der Datentyp "Integer" verwendet, der ganzzahlige Werte speichert, somit entfällt bei der Division eine mögliche Nachkommastelle. Ein Ausschnitt einer erzeugten KML-Datei wird in Abbildung 6.1 dargestellt. Für jede der 20 verschiedenen Geschwindigkeitsstufen (0 - 190 in Zehnerschritten) wird in der KML-Datei ein Element "Style" mit einem "id" Attribut kreiert. Der Wert des "id" Attributs ist die Geschwindigkeitsstufe. Zudem wird innerhalb des "Style" Elements ein "IconStyle" Start- und End-Tag (Etikett, Auszeichner eng. Tag) erzeugt. Zwischen diesen Tags

---

<sup>1</sup>Link zum Schema: <http://www.opengis.net/kml/2.2> (Zugriff: Oktober 2011)

<sup>2</sup>Internetpräsenz der Organisation: <http://www.opengeospatial.org/> (Zugriff: Oktober 2011)

<sup>3</sup>Frei verfügbare Software von Google Inc.: <http://www.google.de/intl/de/earth/index.html> (Zugriff: Oktober 2011)

<sup>4</sup>Microsoft stellt unter der Internetadresse <http://technet.microsoft.com/en-us/library/cc722477.aspx> (Zugriff: Oktober 2011) einige mögliche Befehle vor.

## 6.1. Überprüfung von der FCD-Erhebung

```
<Style id="10"><IconStyle><Icon>
<href>images/10.png</href>
</Icon></IconStyle></Style>
<Placemark>
<name></name>
<styleUrl>#10</styleUrl>
<description>2011-10-01 02:10:39 - 13km/h</description>
<Point>
<coordinates>13.4484310150146,52.5217933654785</coordinates>
</Point>
</Placemark>
```

Abbildung 6.1.: Ausschnitt einer automatisch erzeugten KML-Datei

steht ein "Icon" Element, in dem durch ein "href" Start- und End-Tag auf eine Bilddatei referenziert wird. Innerhalb des "Placemark" Elements wird das Tag "Point" erstellt. Das "Point" Element besitzt als Unterelement "coordinates", in dem als Inhalt der Breiten- und Längengrad durch ein Komma getrennt stehen. Mit dem Unterelement "styleUrl" des "Placemark" Tags wird auf einen zuvor festgelegten Style durch die Identifikationsnummer referenziert. Zudem kann das "Placemark" ein Unterelement namens "description" besitzen, dessen Inhalt in Google Earth angezeigt wird, wenn der Punkt vom Nutzer selektiert wird. In Google Earth können die einzelnen Attribute einer Geoposition zusätzlich zu den gefahrenen Routen betrachtet werden. An dieser Stelle wird das Intervall von Sendungen überprüft und ein Ausschnitt in Abbildung 6.2 gezeigt. Es wurde gezielt ein Ausschnitt gewählt, bei dem



<input checked="" type="checkbox"/>	[keine Name]	2011-10-01 14:30:37 - 0km/h
<input checked="" type="checkbox"/>	[keine Name]	2011-10-01 14:30:42 - 10km/h
<input checked="" type="checkbox"/>	[keine Name]	2011-10-01 14:30:52 - 22km/h
<input checked="" type="checkbox"/>	[keine Name]	2011-10-01 14:31:02 - 36km/h
<input checked="" type="checkbox"/>	[keine Name]	2011-10-01 14:31:12 - 42km/h
<input checked="" type="checkbox"/>	[keine Name]	2011-10-01 14:31:22 - 26km/h

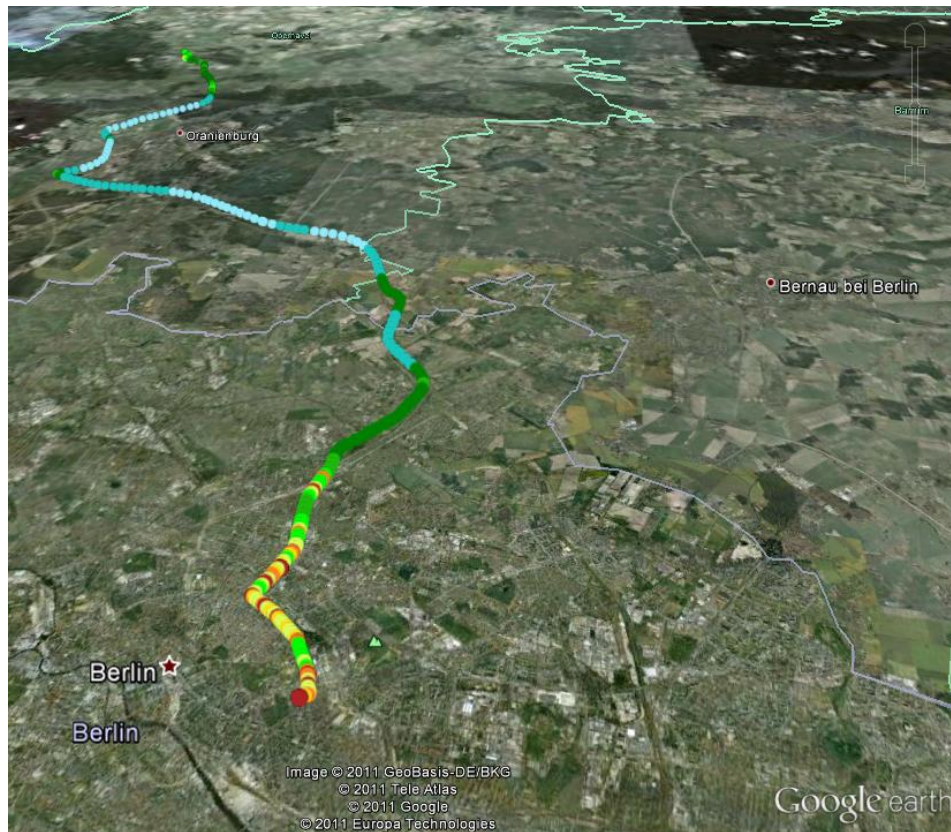
Abbildung 6.2.: Intervall von FCD-Sendungen - Ausschnitt von Google Earth

eine Abweichung hinsichtlich des Intervalls von der FCD-Erfassung gegenüber der angegebenen Sollvorgabe deutlich wird. Aufgrund dessen wird eine Ursachenanalyse im Kapitel 6.1.1 durchgeführt. In Abbildung 6.3 wird die farbliche Zuordnung mittels der gerundeten Geschwindigkeit für eine Geoposition gezeigt und Abbildung 6.4 stellt die Anzeige einer KML-Datei in Google Earth dar.

## 6.1. Überprüfung von der FCD-Erhebung

Farbe	Geschwindigkeit in km/h
●	0 und 10
●	20
●	30
●	40
●	50
●	60 und 70
●	80 und 90
●	100 und 110
●	120 und 130
●	140 und 150
●	160 und 170
●	180 und 190

**Abbildung 6.3.:** Farbliche Darstellung der Geoposition anhand der Geschwindigkeit



**Abbildung 6.4.:** FCD-Sammlung - Route: Berlin Friedrichshain nach Teschendorf (Landkreis Oberhavel) 01.10.2011 14:22 - 15:21 Uhr

### 6.1.1. Ursachenanalyse und Verbesserungsansatz

Entgegen der Erwartung und Annahme wird das in Kapitel 4.1.1 festgelegte Intervall, die Zeitdifferenz zwischen erfassten FCD und bereits gesammelter Information, bei einigen Daten von minimal 10 Sekunden unterschritten. Als Ursache werden folgende Hypothesen aufgestellt:

- Hypothese 1: Das JavaScript FCD-Objekt, in dem alle Positionen nach erfolgreicher Verifizierung einem Array hinzugefügt werden, ist nicht Singleton<sup>5</sup>. Aufgrund dessen schlägt die Verifizierung und Filterung von Geodaten nach der Zeitdifferenz fehl. Bereits gesammelte Positionsdaten würden für die Filterung bei mehreren Objekten eventuell nicht berücksichtigt werden.
- Hypothese 2: Die Filterung und Verifizierung, welche durchgeführt werden, bevor eine neue Geoposition einem Array hinzugefügt wird, sind fehlerhaft. Sind im Array bereits Informationen vorhanden, wird das Intervall zwischen der letzten Position im Array mit der Zeit von aktuell erfassten Ortungsdaten kontrolliert. Sind keine Daten vorhanden, wird die Geoposition ohne zeitliche Überprüfung im Array gespeichert.

Nachfolgend werden eine Überprüfung des Quellcodes und eine Simulation von einigen Geopositionen durchgeführt. Zusätzlich zur Simulation der Ortungsinformationen wird eine Sammlung der Daten als FCD realisiert. Die Untersuchung ergab, dass Hypothese 1 nicht zutrifft. In der Applikation wird die Lokalisierung einmalig erzeugt und initialisiert. Diese Initialisierung findet, wie bereits in Kapitel 4.3.6 erwähnt wurde, in der Datei "geolocation.js" statt. In dieser Datei wird das FCD-Objekt durch die Funktion "Titanium.include()" einmalig eingebunden. Weiterhin ergab die Untersuchung, dass Hypothese 2 zutrifft. Bei der parallelisierten Übertragung und Sammlung von FCD kommt es zu zeitlichen Überschneidungen, die für eine Überprüfung der zeitlichen Intervalle nachteilig sind. Nach erfolgreicher Übertragung zum Server werden vorhandene gespeicherte FCD aus dem Array gelöscht. Wird nun eine neue Geoposition erhalten, wird sie dem Array sofort ohne weitere Überprüfung hinzugefügt. Der erfolgreiche Transfer zum Server impliziert clientseitig eine "Leerung" des Arrays. Obwohl das Intervall zu einer bereits gesendeten

---

<sup>5</sup>Mit Singleton ist ein Entwurfsmuster in der Softwareentwicklung gemeint. Gleichbedeutend mit der Eigenschaft, dass ein Objekt als Einzelstück vorliegt.

## 6.2. Untersuchung der Nutzerakzeptanz

---

und gelöschten Position weniger als 10 Sekunden betragen kann, wird eine neue Position sofort hinzugefügt. Die Ursache des Fehlers wird durch folgenden Verbesserungsansatz korrigiert: Für die Kontrolle einer minimalen Zeitdifferenz von 10 Sekunden wird nicht mehr die letzte Position im Array verwendet. Stattdessen wird die Zeit der letzten hinzugefügten Position global im FCD-Objekt gespeichert. Auf diese Weise wird eine Überprüfung des Zeitintervalls mit der letzten hinzugefügten Position selbst nach Leerung des Arrays durchgeführt.

## 6.2. Untersuchung der Nutzerakzeptanz

Dieser Test beschäftigt sich mit der Akzeptanz der Applikation von Nutzern. Innerhalb der ersten Softwareüberprüfungen durch die Mitarbeiter wurden folgende Hinweise genannt:

- Der Monitor sollte in der Karte angezeigt werden. Auf diese Weise können Verkehrsstauungen während der Fahrt überblickt werden. Andernfalls muss das Gerät durch den Tabwechsel bedient werden.
- Nach einiger Zeit ohne aktive Betätigung schaltete das Gerät in einen Energiesparmodus. Gerade für die Nutzung während der Fahrt ist es essenziell, den Energiesparmodus zu deaktivieren. Ansonsten muss das Gerät kontinuierlich bedient werden, damit die Applikation aktiv auf dem Display bleibt.

Ein derartiges Feedback von den Probanden ist wichtig, um eine positive User Experience zu erzielen. Die Nutzerbemerkungen und -erwartungen wurden berücksichtigt und in der konzeptionellen Phase aufgenommen.

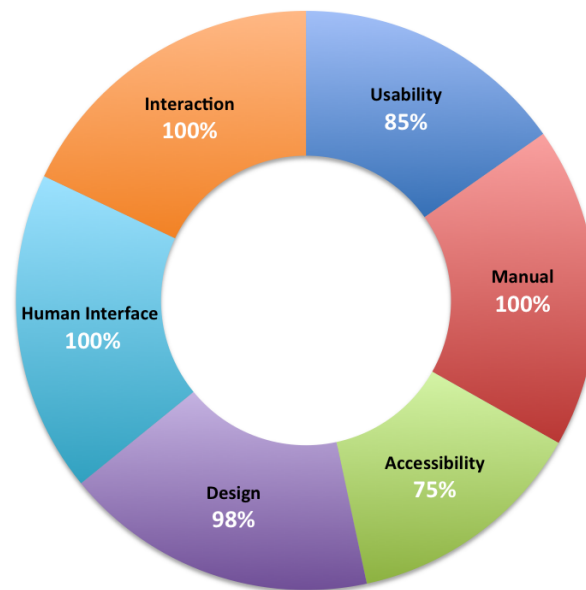
### 6.2.1. Fragenkatalog

Das Konzept der mobilen Applikation (siehe Kapitel 4.1) wurde hinsichtlich der Einflussfaktoren einer positiven User Experience entwickelt. Um die Nutzererfahrung zu messen und zu überprüfen, wird ein Fragenkatalog erstellt, der jeden Faktor gezielt hinterfragt und von Applikationstestern ausgefüllt wird. Aufgrund der bestehenden Anforderungen und Begrenzung an eine geringe Anzahl von Probanden verfügt die Überprüfung und Messung über keine repräsentative Aussagefähigkeit. Diese Durchführung ist als prototypische Umfrage zu verstehen, die nicht für eine

## 6.2. Untersuchung der Nutzerakzeptanz

---

repräsentative Messung eingesetzt wird. Stattdessen wird der Fragenkatalog eingesetzt, um eine grundsätzliche Rückmeldung der prototypischen Umsetzung zu erhalten. Die Umfrage kann nach Genehmigung der DLR-Standortleitung Berlin mit einer deutlich höheren Anzahl an Testpersonen erneut durchgeführt werden. Nachdem die Applikation den Entwicklungsstand des Prototypes erreicht hatte und dieser übergeben wurde, erfolgte zwei Wochen später die Beantwortung von den Probanden. Der Fragenkatalog wird in Anhang D.1 abgebildet. Die Auswertung des Fragenkataloges erfolgt auf eine Art und Weise, bei der für jeden Einflussfaktor ein prozentualer Erfüllungsgrad berechnet wird. Anhand der Probandenbewertung ergeben sich die in Abbildung 6.5 dargestellten Werte. Hierfür wurden die Antworten, die der Punkteskala zugrunde liegen, prozentual gewichtet. Hingegen wurden Antworten, die lediglich positiv oder negativ zu beantworten waren, bei positiver Resonanz mit 100% und andernfalls mit 0% gemessen. Die Bewertung der einzelnen Fragen wird anbei als Excel-Datei<sup>6</sup> auf der Compact Disc (CD) zur Verfügung gestellt.



**Abbildung 6.5.:** Auswertung des Fragenkataloges (n=2)

Die Probanden nannten einige Empfehlungen für eine zukünftige Version:

---

<sup>6</sup>Tabellenkalkulationsprogramm von Microsoft Office

### 6.3. Plattformübergreifende Entwicklung

---

- Die Position des Fahrzeugs auf einer überwachten Route anzeigen.
- Die Möglichkeit, eine Route mit mehr als einem Zwischenpunkt zu definieren.

Weiterhin identifizierten die Applikationstester folgende innovative Funktionalitäten:

- Die Überwachung von gespeicherten Routen und der Abgleich von alternativen Strecken wird möglich.
- Verkehrslageinformationen werden auch auf Nebenstraßen zur Verfügung gestellt.

### 6.3. Plattformübergreifende Entwicklung

Aus der Bewertungsmatrix im Kapitel 3.5 ging hervor, dass die Entwicklung mit dem plattformübergreifenden Framework Titanium Appcelerator erfolgt. Für die Implementierung wurde somit ein Framework verwendet, das die Möglichkeit bietet, nur mit einigen Quellcodeanpassungen eine Entwicklung und somit Portierung für Android zu ermöglichen. Zur Überprüfung dieser Aussage wird eine Ausführung der Applikation mit einem Android Emulator initiiert. Diese Ausführung ist vorerst nicht möglich, da einige iPhone typische Animationen und UI-Elemente verwendet wurden. Dieser Zusammenhang wird in Abbildung 6.6 mit den Fehlermeldungen in der Konsole verdeutlicht.

```
W/iPhone ( 308): (kroll$1: app:///app.js) [11,2732] UI.iPhone.AnimationStyle.FLIP_FROM_RIGHT property isn't supported on Android
I/ActivityManager( 60): Starting activity: Intent { cmp=com.appcelerator.titanium/ti.modules.titanium.ui.TiTabActivity (has extras) }
W/TiTabActivity( 308): (main) [194,2926] Notifying TiTabGroup, activity is created
D/skia ( 308): --- SkImageDecoder::Factory returned null
I/global ( 308): Default buffer size used in BufferedReader constructor. It would be better to be explicit if an 8k-char buffer is required.
D/dalvikvm( 308): GC_FOR_MALLOC freed 3438 objects / 213472 bytes in 71ms
D/KrollContext( 308): (kroll$2: app:///appWindows/map_window.js) [425,3351] Running evaluated script: app:///appWindows/map_window.js
D/dalvikvm( 308): GC_FOR_MALLOC freed 8283 objects / 669696 bytes in 68ms
W/iPhone ( 308): (kroll$2: app:///appWindows/map_window.js) [537,3888] UI.iPhone.SystemButton.REFRESH property isn't supported on Android
W/iPhone ( 308): (kroll$2: app:///appWindows/map_window.js) [7,3895] UI.iPhone.SystemButton.SPINNER property isn't supported on Android
```

**Abbildung 6.6.:** Fehlerhafte Ausführung der App im Android Emulator

Folglich gilt es, typische iPhone-Elemente im Quellcode zu identifizieren und spezifische Alternativen für Android zu implementieren. In der Abbildung 6.7 wird ein Beispiel dargestellt, wie das Smartphone-Betriebssystem festgestellt werden kann und der Style einer Animation entsprechend angepasst wird.



### 6.3. Plattformübergreifende Entwicklung

```
if (Ti.Platform.osname != 'android') {  
    // open the tabgroup and startup screen will be closed with a style  
    tabGroup.open({  
        transition: Titanium.UI.iPhone.AnimationStyle.FLIP_FROM_RIGHT // Animation style  
    });  
}  
else {  
    tabGroup.open();  
}
```

**Abbildung 6.7.:** Titanium Appcelerator OS feststellen

Als weitere Herausforderung zeigt sich, dass die Navigationsoberfläche anders aufzubauen ist, da die Navigationsbar in Android nicht vorhanden ist. Somit sind "rightNavButton" und "leftNavButton" ebenfalls Titanium-Konstrukte, die für eine Android Plattform nicht verfügbar sind. Eine erfolgreiche Durchführung der Applikation mit dem Android Emulator nachdem benötigte Anpassungen vollzogen wurden, wird in Abbildung 6.8 gezeigt.



**Abbildung 6.8.:** Ausführung der App im Android Emulator nach Anpassungen

---

## 7. Ergebnisse und Ausblick

### 7.1. Beschreibung und Bewertung der Vorgehensweise

Zuerst wurden die Anforderungen dieser Arbeit kategorisiert und den Umsetzungsmöglichkeiten nachfolgend gegenübergestellt. Anschließend erfolgte die Erstellung einer Anforderungsmatrix, in der die Fähigkeiten der bestimmten Lösungsalternativen zu den Anforderungen aufgezeigt und grundsätzlich bewertet wurden. Anhand der Anforderungen und deren gewichteten Prioritäten wurden entscheidende Faktoren zur Bewertung der Umsetzungsmöglichkeit gegeben. Es erfolgte eine Beurteilung und Qualifizierung verschiedener Umsetzungsmöglichkeiten. Zugunsten dieser Vorgehensweise wurde die fähigste Umsetzungsvariante in Bezug auf die Anforderungen ausgewählt.

### 7.2. Zusammenfassung der Ergebnisse

Ein wesentliches Ergebnis dieser Arbeit ist die konzipierte und implementierte Verkehrsinformationsapplikation für das iPhone. Die mobile Anwendung wurde als hybride App mittels des plattformübergreifenden Frameworks Titanium entwickelt. Durch das Resultat werden tatsächlich einige Vorteile einer nativen App und einer Web-App vereint und deren Nachteile verringert. Das Framework ermöglicht die Kreation einer performanten nativen iPhone App auf der Basis von Webtechnologien. Die Testphase zeigt auf, dass ein für iPhone implementiertes und funktionierendes Titanium Produkt nicht zwangsläufig ebenfalls für Android sofort lauffähig ist. Es müssen einige programmtechnische Anpassungen getätigt werden. Abschließend gilt festzuhalten, dass die Portierung für Android möglich ist und diese Fähigkeit angesichts einer Ausführung in der Testphase erkennbar wird. Mithilfe der automatisierten Erzeugung einer KML-Datei auf der Basis von gesammelten und gespei-

## 7.2. Zusammenfassung der Ergebnisse

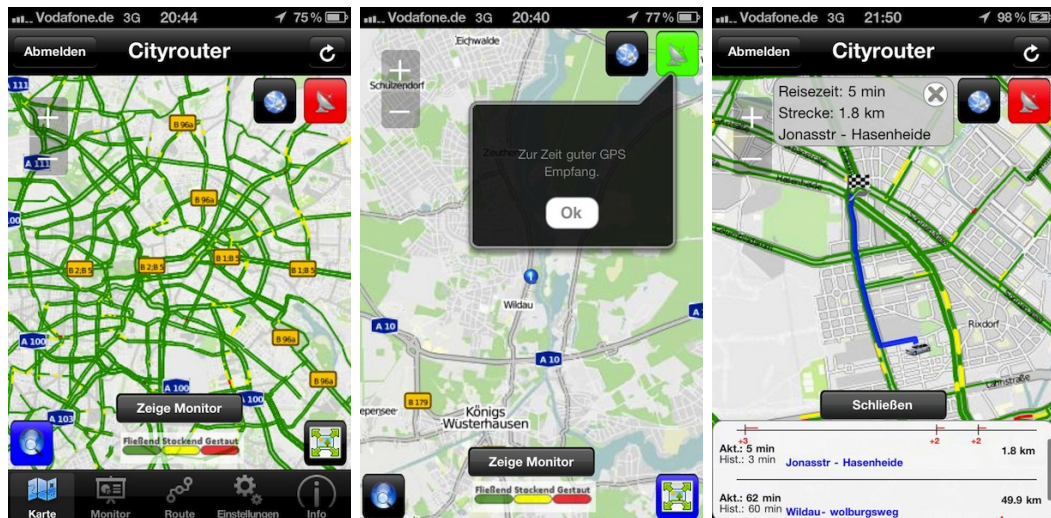
---

cherten FCD, werden die Ergebnisse anschaulich nahegelegt. Die Qualität ist hoch, denn zum einen ist die Datenrate im Vergleich zu üblichen Taxi-FCD bis zu 600% höher und zum anderen wird die Rekonstruktion von gefahrenen Routen deutlich. Eine derartige Überprüfung von gesammelten FCD ermöglichte die Validierung des Sollzustands. Die Daten werden übertragen, sobald eine Verbindung zum Internet und verifizierte FCD vorliegen. Des Weiteren ist eine Modifizierung des festgelegten Zeitabstandes zwischen gesammelten FCD sowie die Einstellung des Intervalls für Sendungen leicht anpassbar. Die erzielten und kontrollierten Resultate werden vom Auftraggeber als "sehr gut" bezeichnet. Ein programmtechnischer Erfolg dieser Arbeit ist die potenzielle Verwendung von Offline-Karten in der prototypischen Applikation. Lokal gespeicherte Kartenbilder ermöglichen eine signifikante Reduzierung der übertragenden Volumen. Die Kartenbilder benötigen mit 11 Zoomstufen nur ein Speichervolumen von 28.1 Megabyte (MB) für Berlin. Durch die Minimierung von Übertragungskapazitäten steht die vorhandene und mögliche Übertragungsrate anderen Funktionen zur Verfügung. Ein weiteres Ergebnis dieser Arbeit ist das mittels Hibernate umgesetzte Datenmodell. Auf Basis der ausgelagerten Konfiguration und des Object-Relational Mapping von Hibernate besteht das Potenzial, das Datenbanksystem flexibel und automatisch zu verändern sowie die Elementstrukturen zu erstellen. Diese Abstraktion ermöglicht eine bedeutende Dynamik für die Entwicklung von Datenbankstrukturen. Das Datenmodell ist demzufolge für unterschiedliche Datenbanksysteme agil einsatzbereit. Um die Nutzererfahrung zu überprüfen, wurde ein Fragenkatalog erstellt, der gezielt jeden Faktor einer User Experience hinterfragt. Der Fragenkatalog wurde von Applikationsprobanden ausgefüllt. Die innovativen Funktionalitäten wurden von den Probanden identifiziert und als Mehrwert erkannt. Ebenfalls erscheint die Applikation klar strukturiert, konsistent, flexibel und einfach in der Steuerung.

### 7.2.1. Die App in Bildern

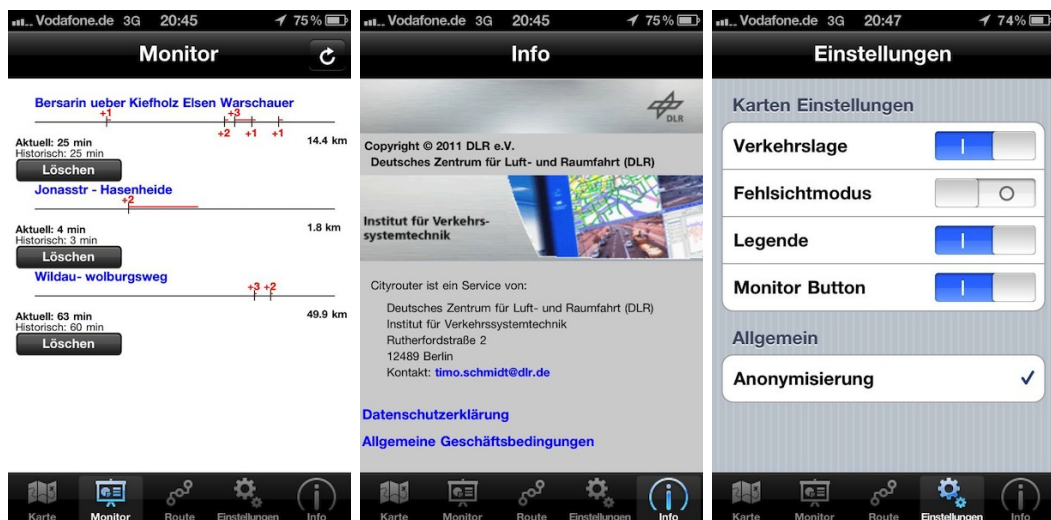
Um das wesentliche Endergebnis dieser Arbeit zu präsentieren, bietet es sich an, die Applikation mit spezifischen Funktionalitäten, Ausführungen und Darstellungen im Folgenden bildlich zu visualisieren.

## 7.2. Zusammenfassung der Ergebnisse



(a) Karten-Tab mit Aktivierung der Lupen-Funktion (Trotz Ortung in Wildau kann die Karte in Berlin betrachtet werden) (b) Vollbildmodus und Auswahl der Ortungsinformation (c) Anzeige des "scrollbaren" Monitors und Auswahl einer gespeicherten Route

Abbildung 7.1.: Die App in Bildern - Das Karten-Tab



(a) Routenüberwachungs-Tab (b) Informations-Tab (c) Einstellungs-Tab

Abbildung 7.2.: Die App in Bildern - Monitor, Info und Einstellungen

### 7.3. Ausblick

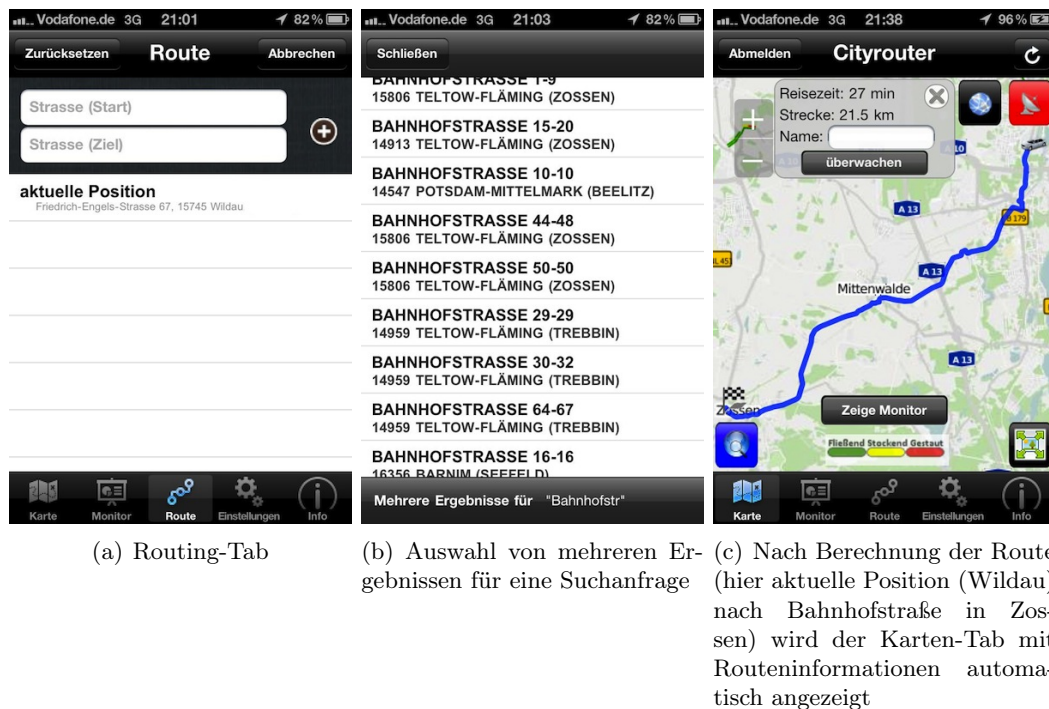


Abbildung 7.3.: Die App in Bildern - Auswahl einer Route

### 7.3. Ausblick

Zukünftige Funktionalitäten und Eigenschaften der mobilen Verkehrsinformationsapplikation wurden in Kapitel 4.1 bereits benannt, um das vollständige Potenzial aufzuzeigen.

Wie bereits in Kapitel 1.4 erwähnt wurde, besteht für diese Arbeit die Annahme des optimalen Falles, bei dem Applikationsnutzer entsprechende Verkehrsteilnehmer für FCD sind. Diese Annahme entspricht aber nicht immer der Realität. Ähnlich wie die festgelegten Zustände von Taxifahrern werden in Zukunft verschiedene Zustände der Applikationsnutzer und adäquate Erkennungsmerkmale definiert. Diese Zustände gilt es bei der FCD-Prozessierung des DLR, der durch die mobilen Applikationen gesammelten FCD, zu identifizieren und unbrauchbare Daten zu verwerfen (beispielsweise von Fußgängern).

Nach Angabe eines betrieblichen Ansprechpartners ist es geplant, dass die anony-

## 7.4. Persönliches Fazit

---

misierter Erhebung von FCD sowie Sicherheitskonzepte innerhalb der nächsten Applikationsversion (Beta-Version) entworfen und umgesetzt werden. Für eine Anonymisierung werden die Identifikationsnummern von Teilnehmern ähnlich wie beim Taxisystem in regelmäßigen Abständen verändert und serverseitig bezogen.

Die Applikation soll laut Aussage des betrieblichen Betreuers in den nächsten Monaten gezielt im Institut und institutsübergreifend an mehrere DLR-Mitarbeiter verbreitet werden. Dieser Schritt kann nur in Absprache und nach Genehmigung von Vorstandsmitgliedern (Gremien) des DLR erfolgen. Hierfür wurde ein Antrag an die Standortleitung Berlin gestellt. Der Prozess der Genehmigung dauert laut Aussage des Betreuers lange an. Die Verteilung der Applikation im Standort ist als zweite Testphase mit deutlich mehr Probanden zu interpretieren. Im Anschluss werden die zu diesem Zeitpunkt vorliegenden Aufnahmekriterien für den App Store evaluiert und eine Erfüllung angestrebt<sup>1</sup>. Ziel ist es, das Produkt nach dieser Phase im App Store anbieten zu können.

## 7.4. Persönliches Fazit

In Kapitel 3.1.1 wurde erläutert, dass für die Ermittlung von Zeitverzögerungen die aktuelle Verkehrslage auf einer Route mit einer "angenommenen" verkehrsgünstigen Lage im Zeitraum von 0-5 Uhr nachts auf der Route verglichen wird. Diese Annahme hat den Vorteil, dass eine "wahre Geschwindigkeit" auf den Strecken zum Abgleich verwendet wird. Oftmals wird eine höhere Geschwindigkeit gefahren, als es laut StVO vorgegeben wird. Die Annahme wurde von mir aber ebenfalls als problematisch identifiziert, da auf vielen Straßen zu den Zeiten 0-5 Uhr aufgrund des Lärmschutzes nur  $30 \frac{km}{h}$  statt tagsüber  $50 \frac{km}{h}$  erlaubt sind. Wird nun tagsüber ein Abgleich durchgeführt, werden keine Verzögerungen festgestellt, obwohl die aktuelle Geschwindigkeit bei stockendem Verkehr auf der Strecke eventuell  $30 \frac{km}{h}$  beträgt. Meine Empfehlung ist, dass auf diesen Straßenabschnitten mit nächtlichem Lärmschutz die am Tage laut StVO vorgegebene Geschwindigkeit anstelle der gefahrenen Geschwindigkeit für einen Vergleich herangezogen wird.

Keine Sollvorgabe, aber persönliche Eigenmotivation war es, Funktionalitäten zu

---

<sup>1</sup>Anforderungen sind in Absatz 3.3 "Program Requirements for Applications" in dem Dokument "iPhone Developer Agreement" von Apple definiert: [https://www.eff.org/files/20100302\\_iphone\\_dev\\_agr.pdf](https://www.eff.org/files/20100302_iphone_dev_agr.pdf) (Zugriff: Oktober 2011)

#### 7.4. Persönliches Fazit

---

kreieren, die wirklich erkennbare Vorteile gegenüber Konkurrenzprodukten schaffen. Hierzu zählt die Steuerung des interessanten Routen-Monitors, Nutzer können Routen in der Applikation speichern und auf Wunsch auch löschen. Erst mit dieser Funktionalität handelt es sich um einen wirklichen Routen-Monitor, da personalisierte Routen überwacht werden. Eine weitere Funktionalität, die die App von anderen abgrenzt und zugleich aus Eigeninteresse erstellt wurde, sind Offline-Karten für Berlin. Die Aufgabenstellung war äußerst umfangreich, vielseitig und bereitete mir viele neue Erkenntnisse und Erfahrungen. Ich möchte mich ganz herzlich bei meinen Hochschulbetreuern Frau Prof. Mohnke und Herrn Prof. Eylert bedanken, die mich durch hilfreiche Hinweise und Feedback unterstützt haben sowie bei Fragen gerne zur Seite standen. Auch bedanke ich mich bei meinem betrieblichen Betreuer Herrn Sohr, der die Weiterentwicklung sowie Verbesserungen der von mir benötigten DLR-Funktionen vorantrieb und mich bei der Masterarbeit unterstützte.

---

## 8. Zusammenfassung

In der heutigen Zeit hat eine effiziente Mobilität sowohl in der Wirtschaft als auch in der Gesellschaft einen hohen Stellenwert. Um den Verkehrsfluss zu optimieren oder Stauungen gezielt zu umfahren, gilt es Verkehrssituationen und -zustände des Straßennetzes zu erfassen. Mit dieser Intention sammelt das Deutsche Zentrum für Luft- und Raumfahrt e.V. (DLR), Standort Berlin, Floating Car Data (FCD). Die FCD-Informationen beinhalten im Verkehrsfluss von Fahrzeugen erhobene geografische Positionen, Geschwindigkeiten, Fahrzeugidentifikation und Zeitpunkt der Ortung. Als Informationsquellen wirken kooperierende Taxizentralen, die Informationen von den Taxis an das DLR weiterleiten. Das FCD-System ist bisher in zweifacher Sachlage begrenzt. Zum einen können Aussagen über Verkehrszustände nur an Stellen getroffen werden, wo das Territorium von Partnern in Großstädten durchfahren wird. Zum anderen ist die Qualität der Prognosen an den prozentualen Ausstattungsgrad und das Intervall von Sendungen gebunden. Diese Arbeit leistet einen bedeutungsvollen Beitrag zur Weiterentwicklung des Systems, indem durch eine Smartphone-Applikation basierend auf das iPhone von Apple, die kontinuierlich FCD an das DLR sendet, vorliegende Begrenzungen aufgehoben werden. Die Entwicklung erfolgte mit dem plattformübergreifenden Framework Titanium des Unternehmens Appcelerator. Nach Faktoren, wie Usability und Design, geht die gezielt konzipierte und insistierte App mit einer positive User Experience (Nutzererfahrung) einher. Die Wirkung der Einflussfaktoren wird durch einen Fragenkatalog überprüft, der durch Applikationsprobanden ausgefüllt wurde. Die App bietet ein grafisches Abbild der Verkehrslage von Berlin, ein verkehrsoptimiertes Routing von einem Start- zum Zielpunkt mit optionalem Zwischenpunkt, eine Routen Überwachung (Routen-Monitor) sowie Offline-Karten für Berlin an. Als innovative Funktionalität und Mehrwert erweist sich der Routen-Monitor. Nutzer können Routen oder alternative Strecken speichern und in der Anwendung vorliegende Verkehrsstörungen betrachten.



---

# Anhang

## A. Informationen über die Beilage (Compact Disc)

Anbei zu dieser Arbeit wird eine CD zur Verfügung gestellt, die folgende Daten enthält:

- Quellcode der Implementierungen
- Audio-Datei (weitere Beschreibung in Anhang C) über die User Experience
- Excel-Datei mit der Auswertung des Fragebogens

Im Dateiordner "Quellcode" befindet sich der Programmcode, der in vier Dateiordnern für die unterschiedlichen Projekte unterteilt ist.

- "CityrouterApp" - Es ist das Projekt der mobilen Applikation.
- "PhoneWebservice" - Hier ist das Projekt des Web Services enthalten.
- "CityrouterHibernate" - In diesem Ordner befindet sich das Projekt zu dem Datenmodell.
- "CityrouterImageLoader" - Hier ist der Quellcode für den Bezug zu Kartendaten enthalten.

### B. Weitere Informationen zum Thema GPS

Im Folgenden werden zusätzlich zu der Ausarbeitung einer Positionsbestimmung mit GPS (Kapitel 2.5.1) für den interessierten Leser weitere Informationen über GPS bereitgestellt. Diese nachträglichen Beschreibungen von GPS sind nicht als Exkurs (in sich abgeschlossene Abschweifung) anzusehen. So wird zum Beispiel der C/A-Code beschrieben, aber der P(Y)-Code (öffentlich nicht verwendbar, hierzu später mehr) nur benannt. Für detaillierte und umfangreiche Beschreibungen von GPS wird die Literatur [Mansfeld 2009] oder [Dodel und Häupler 2009a] sowie die Signal Spezifikation von GPS [USCG 2. Juni 1995] der United States Coast Guard (USCG) empfohlen.

#### B.1. Das Global Positioning System (GPS)

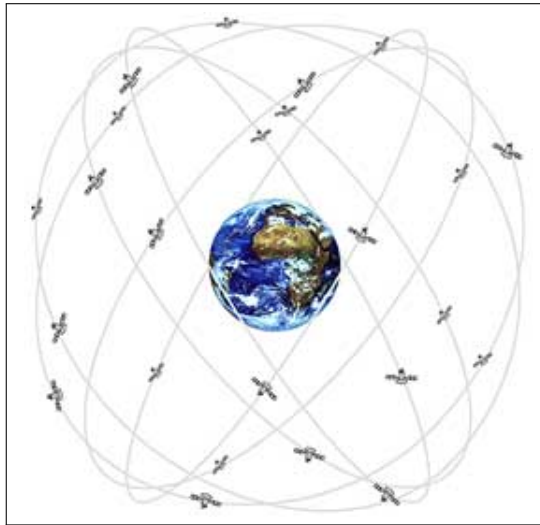
GPS ist ein globales Navigationssatellitensystem, dessen vollständiger Name "Navigational Satellite Timing and Ranging - Global Positioning System (NAVSTAR GPS)" lautet. Es wurde Anfang 1970 ursprünglich nur für militärische Zwecke vom US-Verteidigungsministerium<sup>1</sup> entwickelt. Im Juli 1995 wurde das GPS-System offiziell als voll funktionsfähig<sup>2</sup> eingestuft, mit mindestens 24 Satelliten, die funktionsfähig sind, sich nicht im experimentellen Stadium befinden und in den Umlaufbahnen vorhanden sind [El-Rabbany 2002, S. 1-2]. Es handelt sich um sechs verschiedene Umlaufbahnen, auf denen sich in Abhängigkeit von der aktuellen Konstellation minimal vier Satelliten befinden (siehe Abbildung B1). Der aktuelle Status und die Konstellationen des Satellitensystems sind jederzeit online unter [USCG] einsehbar. Die beim Zugriff (17.07.2011) vorliegende Statusmeldung wird im Anhang B.2 zur Verfügung gestellt. Zu diesem Zeitpunkt wird eine Konstellation von 32 Satelliten deutlich. Die durchschnittliche Entfernung von einem Satelliten zum Mittelpunkt der Erde beträgt 26.560 km, der Erdradius beträgt im Durchschnitt etwa 6.360 km [Langley 1991]. Somit befinden sich die Satelliten in ungefähr 20.200 km<sup>3</sup> Abstand zur Erdoberfläche. Die Umlaufbahnen in dieser Höhe werden als Medium Earth Orbit (MEO) bezeichnet. Jeder Satellit verfügt über hoch genaue Atomuhren. Es

---

<sup>1</sup>Department of Defense (DoD)

<sup>2</sup>Full Operational Capability (FOC)

<sup>3</sup>Es ist die Differenz zwischen der Entfernung von Satelliten zum Mittelpunkt der Erde und dem Erdradius.



**Abbildung B1.:** GPS - Sechs Umlaufbahnen mit je vier Satelliten  
Quelle: [Köhne und Wößner 2009]

sind Rubidium<sup>4</sup> oder Cäsium<sup>5</sup> Atomuhren, die eine Genauigkeit von mindestens die  $10^{-13}$  Sekunde gewähren [Köhne und Wößner 2007] [El-Rabbany 2002, S. 31], also eine Abweichung von einer Sekunde in etwa 300.000 Jahren. In jedem Satelliten befinden sich vier Atomuhren [Zogg 2002, S. 20], sodass ein möglicher Ausfall einer Uhr kompensiert werden kann. Beispielsweise die Satellitentypen [El-Rabbany 2002, S. 4-5] Block II und Block IIA verfügen über zwei Cäsium- und zwei Rubidiumuhren, die Block IIR Typen verfügen nur über Rubidiumuhren [El-Rabbany 2002, S. 31].

### B.1.1. Das Satellitensignal

GPS ist dual im Betrieb und es sind zwei Services PPS (Precise Positioning Service) und SPS (Standard Positioning Service) zu unterscheiden. Der PPS ist für die Nutzung durch das Militär vorgesehen, hierfür wird der P(Y)-Code<sup>6</sup> verwendet und auf Frequenz L2 (1227.6 MHz) ausgesendet<sup>7</sup>. Der SPS dient der zivilen Nutzung und

---

<sup>4</sup>Elementsymbol Rb

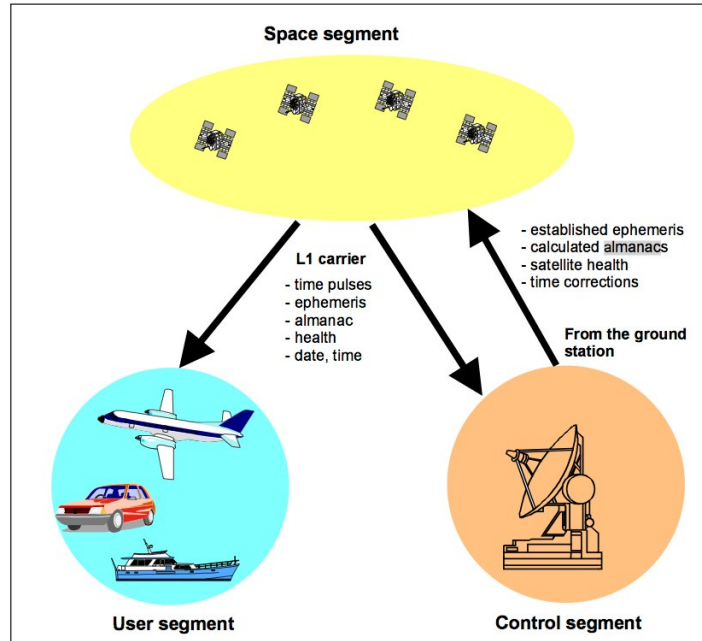
<sup>5</sup>Elementsymbol Cs

<sup>6</sup>Der P (Precision) Code wird durch (Anti Spoofing (AS)) verschlüsselt und dann als P(Y)-Code (Precision/encrypted) bezeichnet.

<sup>7</sup>Öffentlich nicht bekannter Code

## B. Weitere Informationen zum Thema GPS

es wird die Frequenz L1 (1575.42 MHz) eingesetzt [El-Rabbany 2002, S. 9-10]. Die



**Abbildung B2.:** Überblick der Segmente des zivilen (L1) GPS  
Quelle: [Zogg 2002, S. 17]

drei Grundsegmente bei einer zivilen Nutzung und die wesentlichen Informationen, die übertragen werden, sind in Abbildung B2 veranschaulicht. Das Kontrollsegment besteht aus drei Komponenten [USCG 2. Juni 1995, S. 7]:

- Master Control Station - Station der US-Airforce in Colorado, zentrale Kontrolle über das GPS-System
- Monitor Stations
- Ground Antennas

Die Hauptaufgaben des Kontrollsegments besteht beispielsweise darin, die Satelliten zu beobachten und Ephemeridendaten zu berechnen, die Kontrolle der Satellitenuhren und Almanachdaten von allen Satelliten zu vermitteln [Zogg 2002, S. 23]. Bei dem Dienst SPS wird nicht der P(Y)-Code eingesetzt, sondern der C/A-Code (Coarse/Acquisition-Code) [Dodel und Häupler 2009b, S. 181]. Es ist ein PRN-Code

## B. Weitere Informationen zum Thema GPS

---

(Pseudo Random Noise Code<sup>8</sup>), wobei jedem Satelliten ein eindeutiger PRN-Code zugeordnet ist. Es handelt sich um 1023 Bit, die jede Millisekunde wiederholt werden. Die binären Elemente des Codes werden nun im Folgenden als Chips bezeichnet [Braun Juli 2007, S. 17]. Diese Bezeichnung soll den Zusammenhang verdeutlichen, dass ein Chip keine Information im eigentlichen Sinne enthält. Hingegen werden die Binärzeichen aus einem Datenstrom als Bit benannt [Mansfeld 2009, S. 123]. Da es sich um 1023 Chips handelt, bestehen theoretisch  $2^{1023}$  Darstellungsmöglichkeiten dieser Chipfolge. Doch der PRN-Code wird bei GPS nach einer gewissen mathematischen Vorschrift gebildet [El-Rabbany 2002, S. 14]. Es handelt sich um einen Gold-Code<sup>9</sup> [Zogg 2002, S. 22]. Bevor der Gold-Code und die Bildungsvorschrift weiter erläutert werden, wird eine mathematische Klassifizierung des Codes genannt. Für die Erklärung dieser Klassifizierung ist zuerst folgende Definition zu nennen [Eylert 3. Auflage 2010, S. 179-183]:

Eine Menge  $A$  wird als Gruppe bezeichnet, wenn bezüglich der Verknüpfung "Addition"  $\forall a, b$  gilt  $a \in A$  (Abgeschlossenheit) und ebenso das Assoziativgesetz gilt sowie  $\exists$  ein inverses und Null-Element. Gilt dies ebenfalls für die Multiplikation sowie zusätzlich für das Kommutativgesetz und Distributivgesetz bezüglich beider Verknüpfungen, dann ist  $A$  ein Körper.

Gegeben seien die ganzen Zahlen  $a, b, c \in \mathbb{Z}$ , eine natürliche Zahl  $n \in \mathbb{N}$  und es gelte  $a = n * c + b$ . Dann gilt [Eylert 3. Auflage 2010, S. 176]:

$$a \equiv b \bmod n$$

Folglich ergibt die Division von  $a$  und  $b$  mit  $n$  den gleichen Rest. Ist  $n$  eine Primzahl, so bildet die Restklasse<sup>10</sup> ganzer Zahlen modulo  $n$  einen endlichen Körper, den man auch als Galois-Feld  $GF(n)$  bezeichnet [Eylert 3. Auflage 2010, S. 181].

Auf Basis eines veröffentlichten Manuskripts von Robert Gold [Gold Oktober 1967] werden nun weitere Erklärungen zu dem Gold-Code und der Bildungsvorschrift genannt. Der Gold-Code hat bei  $n$ -Elementen eine Länge von  $2^n - 1$  Takten [Mansfeld 2009, S. 123]. Der Code ist die modulo 2 Summe von zwei linearen Mustern G1

---

<sup>8</sup>Es ist ein binärer Code mit einer "scheinbaren" pseudozufälligen Abfolge.

<sup>9</sup>Benannt nach dessen Entwickler Robert Gold.

<sup>10</sup>Die Menge der Restklassen modulo  $m$  besteht aus den Restklassen  $\bar{0}$  bis  $\overline{m-1}$ . Somit besteht für modulo 2 die Menge aus den Restklassen  $\bar{0}$  und  $\bar{1}$  [Eylert 3. Auflage 2010, S. 179].

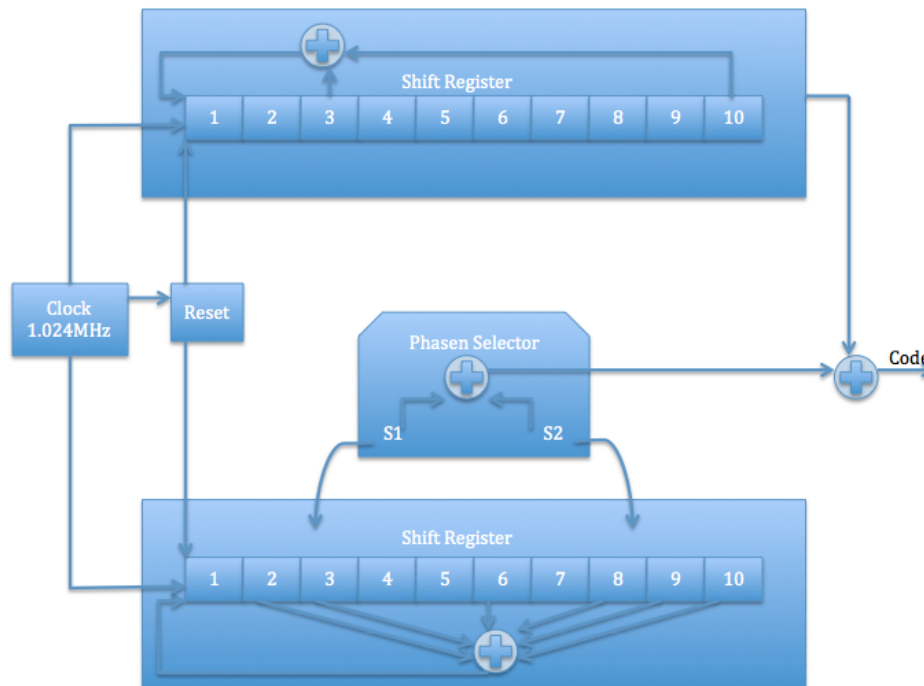
## B. Weitere Informationen zum Thema GPS

und G2, die 1023 Chips lang sind und durch zwei Shift Register Generator mit 10 Elementen (siehe Abbildung B3) gebildet werden. G1 und G2 werden also über dem GF(2) gebildet [Altera Corporation März 2003] und sind als folgende Polynome definiert [USCG 2. Juni 1995, S. 14]:

$$G1: x^{10} + x^3 + 1$$

$$G2: x^{10} + x^9 + x^8 + x^6 + x^3 + x^2 + 1$$

Da der Gold-Code, wie bereits erwähnt, eine Länge von  $2^n - 1$  Takten hat, wird er auch als Code maximaler Länge bezeichnet [Mansfeld 2009, S. 142]. Damit jedem Satelliten ein eindeutiger Gold-Code zugeordnet ist, werden von dem G2 Polynom für jeden Satelliten zwei unterschiedlich festgelegte Code-Phasen selektiert, die per modulo 2 addiert werden. Ebenfalls wird für jeden Satelliten eine unterschiedliche Chip-Verschiebung (eng. delay) zwischen 5 - 950 angegeben [USCG 2. Juni 1995, S. 15], die beim G2 durchgeführt wird [Mansfeld 2009, S. 124].



**Abbildung B3.:** Skizze des Code-Generators  
In Anlehnung an [USCG 2. Juni 1995, S. 17]

## B. Weitere Informationen zum Thema GPS

Diese definierten und auf einen Satelliten referenzierten Code-Phasen werden in Abbildung B4 gezeigt. Die Satelliten Identifikationsnummer wird ebenso als "PRN-Nummer" bezeichnet [Braun Juli 2007, S. 14]. C/A-Code 34 und 37 sind identisch, demzufolge werden 36 unterschiedliche Codes definiert. Die Sequenzen 33 bis 37 sind für einen anderen Nutzen reserviert und werden von den Satelliten nicht verwendet. Liegt dem GPS-Empfänger diese PRN-Nummer zu einem empfangenen

Satellite ID Number	GPS PRN Signal Number	Code Phase Selection	Code Delay Chips	First 10 Chips Octal* C/A
		C/A (G2)	C/A	
1	1	2 ⊕ 6	5	1440
2	2	3 ⊕ 7	6	1620
3	3	4 ⊕ 8	7	1710
4	4	5 ⊕ 9	8	1744
5	5	1 ⊕ 9	17	1133
6	6	2 ⊕ 10	18	1455
7	7	1 ⊕ 8	139	1131
8	8	2 ⊕ 9	140	1454
9	9	3 ⊕ 10	141	1626
10	10	2 ⊕ 3	251	1504
11	11	3 ⊕ 4	252	1642
12	12	5 ⊕ 6	254	1750
13	13	6 ⊕ 7	255	1764
14	14	7 ⊕ 8	256	1772
15	15	8 ⊕ 9	257	1775
16	16	9 ⊕ 10	258	1776
17	17	1 ⊕ 4	469	1156
18	18	2 ⊕ 5	470	1467
19	19	3 ⊕ 6	471	1633
20	20	4 ⊕ 7	472	1715
21	21	5 ⊕ 8	473	1746
22	22	6 ⊕ 9	474	1763
23	23	1 ⊕ 3	509	1063
24	24	4 ⊕ 6	512	1706
25	25	5 ⊕ 7	513	1743
26	26	6 ⊕ 8	514	1761
27	27	7 ⊕ 9	515	1770
28	28	8 ⊕ 10	516	1774
29	29	1 ⊕ 6	859	1127
30	30	2 ⊕ 7	860	1453
31	31	3 ⊕ 8	861	1625
32	32	4 ⊕ 9	862	1712
***	33	5 ⊕ 10	863	1745
***	34**	4 ⊕ 10	950	1713
***	35	1 ⊕ 7	947	1134
***	36	2 ⊕ 8	948	1456
***	37**	4 ⊕ 10	950	1713

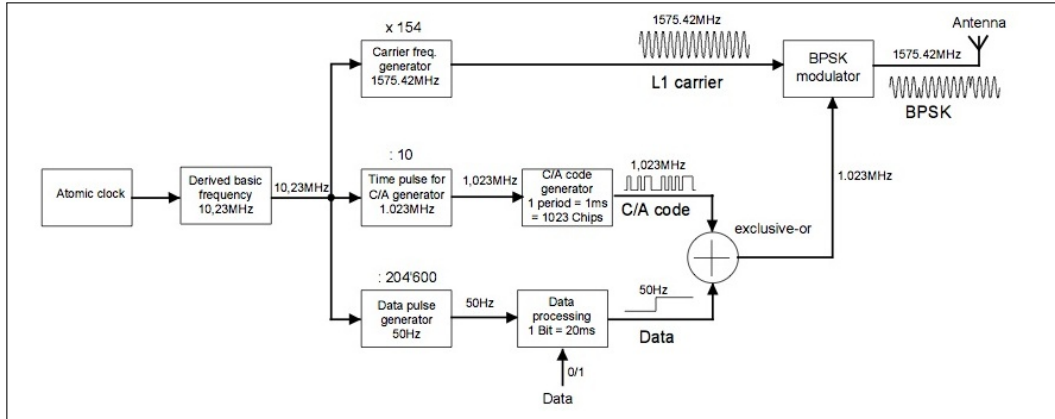
**Abbildung B4.:** Code-Phasen und Chip-Verschiebung

Quelle: [USCG 2. Juni 1995, S. 15]

Satelliten vor, ist die Erzeugung einer identischen Gold-Code-Referenz anhand der zugewiesenen Code-Phasen sowie der angegebenen Chip-Verschiebung möglich. Der C/A-Code ist somit nicht pseudozufällig, sondern reproduzierbar. Deshalb wird der Code als pseudostatistisch bezeichnet [Mansfeld 2009, S. 142]. Durch

## B. Weitere Informationen zum Thema GPS

die Verwendung von 10 Elementen beim Shift Register existieren 1023 Takte, die auch als Chipbreite bezeichnet werden [Braun Juli 2007, S. 17]. Die 1023 Chips werden, wie bereits erwähnt jede Millisekunde wiederholt, aufgrund dessen hat der C/A-Code eine Chiprate von  $1,023 \cdot 10^6$  chips/s [El-Rabbany 2002, S. 14]. In Abbildung B5 wird gezeigt, wie die Daten und Codes für die zivile Nutzung



**Abbildung B5.:** Darstellung von GPS Signalerzeugung und Modulation

Quelle: [Zogg 2002, S. 22]

erzeugt und vom Satelliten ausgesendet werden. Die Grundfrequenz  $f_1$  ist gleich 10.23 MHz und wird im Satelliten von einer der vier mitschwingenden Atomuhren abgeleitet. Weiterhin ergibt sich die Periodendauer (Grundtaktdauer) nach [Eylert 3. Auflage 2010, S. 82]:

$$f_1 = \frac{1}{T_P}$$

Demzufolge ist die Periodendauer  $T_P = \frac{1}{10,23 \cdot 10^6}$  Sekunden.

Unter der Verwendung des Code Division Multiple Access (CDMA) senden alle Satelliten auf der gleichen Frequenz [Kaplan und Hegarty 2006, S. 3-5]. Besonders wichtig ist eine Optimierung der PRN-Codes bezüglich des Korrelationsverhaltens. Um untereinander Störungen zu vermeiden, soll die Kreuzkorrelation von zwei unterschiedlichen PRN-Codes eine Nullfolge ergeben [Fabig 1995, S. 39].

Bei der Signalanalyse ergibt sich diese Forderung durch die Formel der idea-



## B. Weitere Informationen zum Thema GPS

---

len Kreuzkorrelation nach [Mansfeld 2009, S. 143], wobei  $i \neq k$ :

$$C_{i,k}(\tau) = \int_{-\infty}^{+\infty} (PRN)_i(t)(PRN)_k(t + \tau)d\tau = 0$$

$\tau$ : Verschiebung der Code-Phase

$PRN$ : PRN-Code von den Satelliten  $i$  und  $k$

Diese ideale Forderung ist in der Praxis aber nur bedingt möglich [Mansfeld 2009, S. 143]. Stattdessen wird aber eine geringe Kreuzkorrelation erzielt, sodass verschiedene Codes leicht zu unterscheiden sind. Weiterhin besteht ein eindeutiges und sehr ausgeprägtes Maximum durch eine hohe MSR (Main to Side Maximum Ratio)<sup>11</sup> bei der Autokorrelation. Hier werden zwei gleiche Binärfolgen korreliert und bei der Korrelation liegt der höchste Wert vor, wenn keine Verschiebung besteht [Fabig 1995, S. 90-92]. Laut [Mansfeld 2009, S. 142] gilt für einen Code maximaler Länge mit der Periodendauer (oder Breite)  $T$  die Autokorrelation:

$$C(\tau) = \frac{1}{NT} \int_0^{NT} (PRN)(t)(PRN)(t + \tau)d\tau$$

Liegt bei dem Referenzsignal keine Verschiebung vor (also  $\tau = 0$ ), hat die Autokorrelationsfunktion den Maximalwert  $C(0) = 1$ . Die Korrelationsfunktionen beschreiben die Ähnlichkeit eines Signals zu einem um  $\tau$  zeitverschobenen Signal. Der C/A-Code wird mit dem Datenstrom per Exklusiv-Oder Operation (XOR)<sup>12</sup> verknüpft. Es ergibt sich ein Spreizcode, da der Nutzdatenstrom über eine geringere Bandbreite<sup>13</sup> als die PRN-Sequenz verfügt und somit in seinem Spektrum gespreizt wird [Mansfeld 2009, S. 121]. Der Vorgang der Signalspreizung wird auch als Direct Sequence Spread Modulation (DSSS) bezeichnet. Die erhaltenen Ausgangsdaten werden mit dem L1 Frequenzträger durch Bipolar-Phase Shift Key (BPSK) moduliert. Es ist eine Phasenumtastung<sup>14</sup>, bei der die Trägerfrequenz zwei Phasenlagen  $0^\circ$  und  $180^\circ$

---

<sup>11</sup>Das Maximum im Verhältnis zum Betrag des größten Nebenmaximums

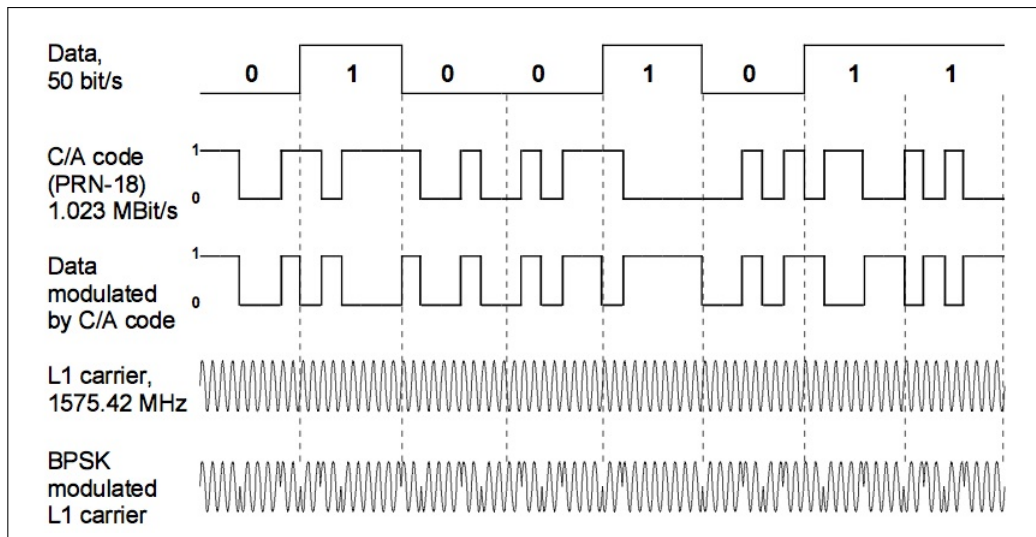
<sup>12</sup>Es ist die "exklusiv-Oder" Verknüpfung. Liegen 2 binäre Eingänge vor, die verknüpft werden, ergibt sich immer dort im Ausgang der Status 1, wo bei den Eingängen ein unterschiedlicher Status existiert. Andernfalls wird der binäre Status 0 angegeben.

<sup>13</sup>Gleich dem Verhältnis zwischen Bit und Chip-Rate

<sup>14</sup>Phase Shift Keying (PSK)

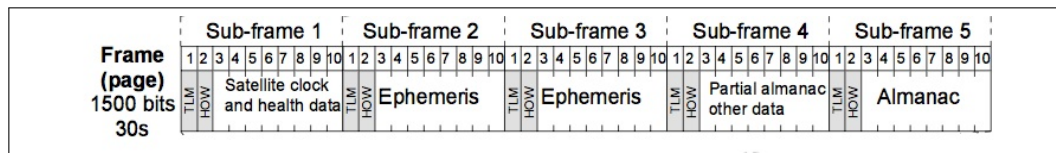
## B. Weitere Informationen zum Thema GPS

(im Bogenmaß  $\pi$ ) annehmen kann (entspricht dem binären Status 0 und 1). Jede Veränderung in den modulierten Daten führt zu einer  $180^\circ$  Veränderung in der L1 Trägerfrequenz. Die Signalstrukturen und Phasenumtastung wird in Abbildung B6 verdeutlicht. Die Struktur eines Datenframes wird in Abbildung B7 dargestellt. Die



**Abbildung B6.:** Struktur der Signale und deren Modulation

Quelle: [Zogg 2002, S. 21]



**Abbildung B7.:** Aufbau eines Datenframes

Quelle: [Zogg 2002, S. 26]

Datenfrequenz beträgt 50 Hz und die Bitrate somit 50 Bit pro Sekunde, sodass die 1500 Bit eines Datenframes innerhalb von 30 Sekunden gesendet werden können [Mansfeld 2009, S. 122]. Der Frame wird in fünf Subframes unterteilt, die innerhalb von sechs Sekunden übermittelt werden können. Die GPS-Navigationsnachricht besteht aus 25 Datenframes, die insgesamt 37.500 Bit groß sind und eine Übertragungszeit von 750 Sekunden, also 12,5 Minuten benötigen [El-Rabbany 2002, S. 15]. Die Navigationsnachricht beinhaltet wesentliche Informationen, wie:

## B. Weitere Informationen zum Thema GPS

---

- Ephemeriden - genaue Beschreibung der Umlaufbahn des sendenden Satelliten
- Almanach - die Beschreibung der Umlaufbahnen von allen Satelliten
- Ionosphäre Informationen - unterstützen die Korrektur der Signallaufzeit
- Satellitenzeit - genaue Zeit des übertragenden Satelliten
- Satellitenzustand - Status des sendenden Satelliten, Verwertbarkeit der Daten (Gültigkeit der Daten)

### B.1.2. Ausbreitungsgeschwindigkeit des Satellitensignals in der Atmosphäre

Wie bereits in Kapitel 2.5.1 erläutert wurde, breitet sich das Signal des Satelliten in dem Vakuum des Weltalls mit Lichtgeschwindigkeit ( $299.792.458 \frac{m}{s}$ ) aus. In der Atmosphäre kann die Geschwindigkeit unterschiedlich sein. Die Atmosphäre wird in verschiedene Schichten eingeteilt: Troposphäre, Stratosphäre, Mesosphäre und Thermosphäre. Die Thermosphäre absorbiert energiereiche kosmische Strahlung, hierbei entstehen Ionen und Elektronen<sup>15</sup>. Deshalb wird die Thermosphäre ebenso als Ionosphäre [Paeger 2007] bezeichnet, sie befindet sich etwa in der Höhe zwischen 50-1000 km [El-Rabbany 2002, S. 36]. Die Ausbreitungsgeschwindigkeit des Satellitensignals in der Erdatmosphäre ist von dem Anteil der elektrisch geladenen Teilchen in der Ionosphäre abhängig. Diese variieren in Abhängigkeit von der Höhe und der chemischen Konstellation. Die Atmosphäre ist nicht in einem chemischen Gleichgewicht. Die Zusammensetzung der Stoffe kann an verschiedenen Regionen in gleicher Höhe unterschiedlich sein und sich im Laufe der Zeit verändern. Zudem ist die Eindringtiefe der solaren Strahlung abhängig von der Wellenlänge, da eine Wellenlänge von über 400 nm fast nicht mehr absorbiert wird [Veckenstedt 2006]. Die Geschwindigkeit des Satellitensignals kann aufgrund dieser Zusammensetzung beim Durchströmen der Atmosphäre variieren. Signale können von geladenen Teilchen gebrochen werden, was die Signalrichtung und -geschwindigkeit beeinflusst. Hierdurch kann eine gemessene Position um bis zu vier Meter abweichen [Zogg 2002, S. 35].

---

<sup>15</sup>Energereiche UV-Strahlung trifft bei der Atmosphäre auf Luftmoleküle, die dadurch gespalten und ionisiert werden. Hierdurch entstehende Atome können verschiedene Ladungen (positiv oder negativ) aufweisen [Veckenstedt 2006].

## B. Weitere Informationen zum Thema GPS

### B.1.3. Signalempfang und -auswertung des GPS-Empfängers

Die Ephemeriden-Daten enthalten die genaue Position von dem sendenden Satelliten, ein Ausschnitt der Information wird in Abbildung B8 dargestellt. Der erste

PG01 -11044.805800 -10475.672350 21929.418200 189.163300
--

**Abbildung B8.:** Ausschnitt der Ephemeriden-Daten eines The International GNSS Service (IGS)

<ftp://igscb.jpl.nasa.gov/pub/data/format/sp3c.txt> (Zugriff: 26. Oktober 2011)

Wert steht für die PRN-Kennung des Satelliten. Die nächsten drei Parameter sind die x-, y- und z-Koordinaten des Satelliten in Kilometer. An der fünften Stelle steht die Abweichung der Satellitenuhr in Mikrosekunden.

Die Almanach-Daten ermöglichen eine Berechnung der Bahndaten und groben Positionen aller Satelliten und somit eine Eingrenzung des Bereichs, in dem gesucht werden muss. In Abbildung B9 wird ein Ausschnitt der Almanach für zwei Satelli-

```
***** Week 635 almanac for PRN-01 *****
ID: 01
Health: 000
Eccentricity: 0.4358291626E-003
Time of Applicability(s): 319488.0000
Orbital Inclination(rad): 0.9604840940
Rate of Right Ascen(r/s): -0.7703178011E-008
SQRT(A) (m 1/2): 5153.617676
Right Ascen at Week(rad): 0.2648420452E+001
Argument of Perigee(rad): 1.397181235
Mean Anom(rad): 0.1272893213E+000
Af0(s): 0.5435943604E-004
Af1(s/s): 0.2546585165E-010
week: 635

***** Week 635 almanac for PRN-02 *****
ID: 02
Health: 000
Eccentricity: 0.1056957245E-001
Time of Applicability(s): 319488.0000
Orbital Inclination(rad): 0.9390622920
Rate of Right Ascen(r/s): -0.7966046104E-008
SQRT(A) (m 1/2): 5153.536621
Right Ascen at Week(rad): 0.2639326298E+001
Argument of Perigee(rad): -2.920123428
Mean Anom(rad): 0.2285483444E+001
Af0(s): 0.3623962402E-003
Af1(s/s): 0.0000000000E+000
week: 635
```

**Abbildung B9.:** Ausschnitt der Almanach  
Quelle: [USCG]

ten dargestellt. Diese Informationen stammen aus der 635ten Woche seit dem 22.

## B. Weitere Informationen zum Thema GPS

---

August 1999, die am 24. Oktober 2011 begann. Der erste Wert ist die PRN-Nummer des Satelliten. "Health" gibt die Funktionsfähigkeit an und beträgt 000, wenn der Satellit funktionsfähig ist. "Eccentricity" gibt die Abweichung der Satellitenbahnen von einer Kreisform an. "Time of Applicability" ist die Zeit in Sekunden, in welcher der Almanach in der Umlaufbahn erstellt wurde. "Orbital Inclination" bestimmt die Neigung der Satellitenbahn zum Äquator. Die Wurzel aus der großen Halbachse "SQRT(A)" steht für den Bahnradius. Bei PRN-1 beträgt er  $5153,617676^2$  m, also etwa 26559 km und der Satellit befindet sich somit aus Perspektive der Erdoberfläche in ungefähr 20199 km Höhe. Der Punkt, an dem der Satellit die Äquatorebene aufsteigend<sup>16</sup> schneidet, wird im Folgenden als "Aufstiegspunkt" benannt. "Right Ascension at Week" ist der Winkel, der sich vom Erdmittelpunkt (Scheitelpunkt des Winkels) zu dem Frühlingspunkt (vernal equinox)<sup>17</sup> und Aufstiegspunkt ergibt. "Argument of Perigee" steht für den Winkel, der vom Erdmittelpunkt aus gesehen zum Aufstiegspunkt und Perigäum<sup>18</sup> bemessen wird. Nachdem nun die Bahn des Satelliten durch diese genannten Werte definiert ist, gibt "Mean Anom" (mittlere Anomalie) an, wo sich der Satellit auf dieser Umlaufbahn etwa befindet. Hat der Satellit das Perigäum verlassen und bewegt sich zum Apogäum, ist die mittlere Anomalie positiv, andernfalls negativ (der Wert  $g$  ist in Grad  $-180 \geq g \leq 180$ ).

Im Empfänger wird zeitgleich zur Satellitenaussendung des Codes zu jedem empfangenen PRN-Code eine Referenzimpulsfolge anhand der PRN-Kennung erzeugt [Braun Juli 2007, S. 14] [Mansfeld 2009, S. 138]. Ebenfalls wird eine identische Frequenz durch einen in Phase und Frequenz nachsteuerbaren Empfänger-Oszillator erstellt [Prinz 2009] sowie der Referenzimpuls auf diese Frequenz moduliert. Auf diese Weise ist, wie in Abbildung B.2 dargestellt, die Signallaufzeit ermittelbar. Schrittweise wird der gesendete PRN-Code mit dem Referenz-Code korreliert, also die binären Elemente werden einzeln multipliziert. Die Summe dieser Multiplikationen nimmt den höchsten Wert an, wenn die Codes zeitlich nicht verschoben sind [Mansfeld 2009, S. 138]. Es wird bei jedem Schritt der Referenz-Code um ein Code-Element (Chip) verschoben, bis der höchste Summenwert erkannt wird und so die

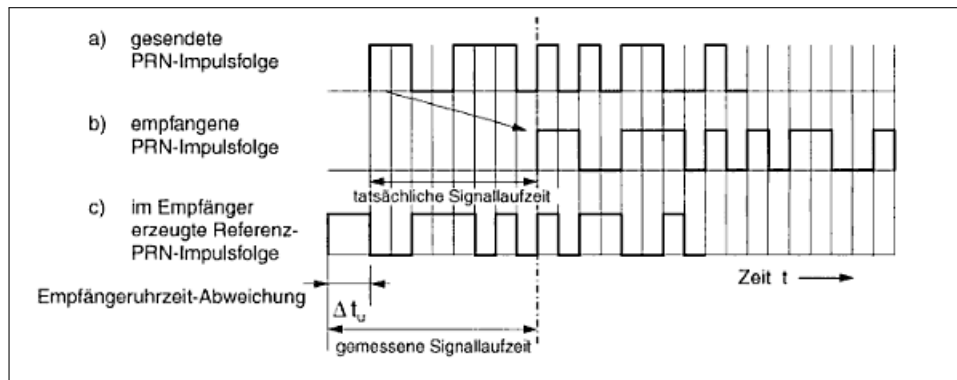
---

<sup>16</sup>Von südlicher nach nördlicher Hemisphäre

<sup>17</sup>Es ist der Schnittpunkt der Ekliptik, die angenommene Sonnenumlaufbahn eines Jahres, mit dem (in den Himmel projizierten) Äquator. Die Sonne erreicht den Frühlingspunkt etwa am 23. März.

<sup>18</sup>Es ist der Punkt, bei dem der Satellit den geringsten Abstand zur Erde hat und der Punkt wird auch als Erdnähe bezeichnet. Hingegen wird die Erdferne als Apogäum benannt.

## B. Weitere Informationen zum Thema GPS



Quelle: [Mansfeld 2009, S. 138]

Codes zeitlich übereinstimmen. Auf diese Weise wird folglich die Signallaufzeit festgestellt. Wird das empfangene Satellitensignal, das die Verbindung zwischen C/A-Code und Datenstrom enthält, mit dem in Phase übereinstimmenden Referenz-Code verknüpft, können die Nutzdaten gewonnen werden [Prinz 2009].

## B. Weitere Informationen zum Thema GPS

### B.2. GPS Status Meldung vom 17.07.2011

```
UNCLASSIFIED
GPS OPERATIONAL ADVISORY      198.0A1
SUBJ: GPS STATUS      17 JUL 2011

1. SATELLITES, PLANES, AND CLOCKS (CS=CESIUM RB=RUBIDIUM):
A. BLOCK I : NONE
B. BLOCK II: PRNS 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
  PLANE : SLOT B6, D6, D1, C2, D4, E6, C6, A4, A3, A1, E6, D2, B4, F3
  CLOCK :      RB, CS, RB, CS, RB, RB, RB, RB, CS, CS, CS, RB, RB, RB
  BLOCK II: PRNS 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
  PLANE : SLOT F1, F2, B1, C4, E4, C3, E1, D3, E2, F4, D5, B2, F5, A6
  CLOCK :      RB, RB, RB, RB, RB, RB, RB, RB, RB, RB, CS, RB, RB, CS
  BLOCK II: PRNS 28, 29, 30, 31, 32
  PLANE : SLOT B3, C1, B5, A2, E5
  CLOCK :      RB, RB, CS, RB, RB

2. CURRENT ADVISORIES AND FORECASTS :
A. FORECASTS:      FOR SEVEN DAYS AFTER EVENT CONCLUDES.
NANU      MSG DATE/TIME      PRN TYPE      SUMMARY (JDAY/ZULU TIME START - STOP)

B. ADVISORIES:
NANU      MSG DATE/TIME      PRN TYPE      SUMMARY (JDAY/ZULU TIME START - STOP)

2011042      132058Z May 2011      30      UNUSUFN      133/2055-/
C. GENERAL:
NANU      MSG DATE/TIME      PRN TYPE      SUMMARY (JDAY/ZULU TIME START - STOP)

2011045      222019Z Jun 2011      GENERAL      /-/
2011046      122307Z Jul 2011      GENERAL      /-/
2011047      160718Z Jul 2011      01      LAUNCH      /-/

3. REMARKS:
A. THE POINT OF CONTACT FOR GPS MILITARY OPERATIONAL SUPPORT IS THE GPS
OPERATIONS CENTER AT 719-567-2541 OR DSN 560-2541.
B. CIVILIAN: FOR INFORMATION, CONTACT US COAST GUARD NAVCEN AT
COMMERCIAL 719-567-2541 24 HOURS DAILY AND INTERNET
HTTPS://WWW.NAVCEN.USCG.GOV
C. MILITARY SUPPORT WEBPAGES CAN BE FOUND AT THE FOLLOWING
HTTPS://gps.afspc.af.mil/GPS OR HTTPS://gps.afspc.af.mil/GPSOC
```

Quelle:[USCG]

### C. Human Interface und User Experience

Auf der CD, die dieser Masterarbeit anhängt, befindet sich eine Audiodatei. Alternativ ist die Datei unter [http://www.adaptivepath.com/uploads/archive/media/ap-interview-don\\_norman-peterme.mp3](http://www.adaptivepath.com/uploads/archive/media/ap-interview-don_norman-peterme.mp3) (Zugriffsdatum 1. August 2011) verfügbar. Hierbei handelt es sich um die Konversation zwischen Donald Norman (Schriftsteller, ehemaliger Professor für Kognitionswissenschaften der University of California, San Diego und für Informatik an der Northwestern University, ehemals Vizepräsident der Advanced Technology Group bei Apple) und Peter Merholz (Mitgründer von Adaptive Path). Die Konversation wurde am 13. Dezember 2007 veröffentlicht, sie ist ein "Vorgeschmack" (Vorgespräch) zu den Dialogen, die auf der UX Week Conference 2008 geführt worden seien. In Anlehnung an [Jordans Oktober 2008, Vgl.] werden folgende markante Stellen empfohlen:

- (1) 8:10 Minute - Peter Merholz liest eine E-Mail von Donald Norman vor (etwa 1998 geschrieben), in der seine Motivation zur Definition von User Experience im Jahr 1993 genannt wird.
- 9:42-11:27 - Donald Norman beantwortet die Frage im Zusammenhang mit der Evolution von Gedanken und Definitionen. Wenn eine Begriffsbestimmung über die Zeit nicht ihre ursprüngliche Bedeutung verliert, sondern positiv erweitert wird, sei es eine gelungene Evolution.
- (2) 11:28-12:01 - Donald Norman mahnt davor, dass bei der User Experience teilweise eine negative zeitliche Evolution stattfand und seine eigentliche Bedeutung verloren ging. Die User Experience würde von Einigen in einem falschen Kontext verwendet werden.



## D. Testphase

---

### D. Testphase

#### D.1. Fragenkatalog

(1. Usability)(1.1 Effektivität) Haben Sie das Gefühl, dass Sie mit der Software Ihre (Fahr-)Ziele effektiver erreichen können?	<input type="checkbox"/> Ja <input type="checkbox"/> Nein, da _____
(1.2 Effizienz) Wie empfinden Sie die Geschwindigkeit der Software?	<b>Skala:</b> 10 Punkte (sehr schnell) bis 0 Punkte (Sehr langsam) _____
(1.3 Zuverlässig) Wie zuverlässig ist das Programm Ihrer Erfahrung nach?	<b>Skala:</b> 10 Punkte bis 0 Punkte _____
Vertrauen Sie der Software und dem Anbieter?	<input type="checkbox"/> Ja <input type="checkbox"/> Nein, begründet: _____
(1.4 Funktionalität) Finden Sie die Software neuartig, innovativ?	<input type="checkbox"/> Ja, innovativ ist <input type="checkbox"/> Nein, da _____
Welche Funktionalität fehlt der Software, die Sie benötigen oder gerne nutzen würden?	_____ _____
(2. Interaktion)(2.1 Feedback) Wissen/Sehen Sie ob Aktionen der Software erfolgreich waren? Erhalten Sie Informationen /Ergebnisse?	<input type="checkbox"/> Ja <input type="checkbox"/> Nein, beispielsweise: _____
(2.2 Flexibilität) Können Sie die Software flexibel anpassen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nein, beispielsweise: _____
(3. Barrierefreiheit) Wie bewerten sie die Barrierefreiheit der Applikation?	<b>Skala:</b> 10 Punkte (sehr gut) bis 0 Punkte (Sehr schlecht) _____
(4. Intuitivität/ Human-Interface) Ist die Steuerung der Applikation für Sie verständlich und einfach?	<input type="checkbox"/> Ja <input type="checkbox"/> Nein, Grund: _____
(5. Design) (5.1 Konsistenz) Sind die Elemente in der Software eindeutig?	<input type="checkbox"/> Ja <input type="checkbox"/> Nein, beispielsweise _____
Verwendet die Software Ihnen bekannte darstellende Elemente und Konventionen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
(5.2 Gestaltung) Wie übersichtlich finden Sie die Struktur der Applikation?	<b>Skala:</b> 10 Punkte (sehr gut) bis 0 Punkte (Sehr schlecht) _____
(5.3 Anordnung Oberfläche) Ist die Anordnung der Elemente logisch?	<input type="checkbox"/> Ja <input type="checkbox"/> Nein, beispielsweise _____
(6. Manual) Erhalten sie Informationen über den Anbieter? Besteht die Möglichkeit bei Bedarf Kommunikation herzustellen?	<input type="checkbox"/> Ja <input type="checkbox"/> Nein, da _____

## D. Testphase

### D.2. Batch Skript für die automatische Erzeugung einer KML-Datei

```
1 @ECHO OFF
2 CD D:\Tracking\KML
3
4 REM ***** Host and Port of the database for FCD *****
5 SET HOST=129.247.218.10
6 SET PORT=5432
7
8 IF "%2%"==" " GOTO INPUT
9 SET VEH_ID=%1
10 SET DATE=%2
11 GOTO CALC
12 REM ***** Get the id and date by user input *****
13 :INPUT
14 SET /P VEH_ID=Vehicle id:
15 SET /P DATE=date[YYYYmmdd]:
16 :CALC
17 SET FILE=VEH_ID%DATE%.kml
18 echo %FILE% selecting tracking positions to %FILE%
19
20 REM ***** Create the KML file and calculate the gtrac for each Geodata for given Speed *****
21 "C:\Programme\PostgreSQL\9.0\bin\psql" -h%HOST% -d%DB% -u%USER% -c "select '<Placemark><name></name><styleurl>#</styleurl>' || ('<speed>
22 <speed>' || speed || '</speed>' || 'km/h</description><point><coordinates>' ||
23 st_x(st_astext(position)) || ',' || st_y(st_astext(position)) || '</coordinates></Point></Placemark>' FROM fcd_mobile Where user_id=VEH_ID% and
24 to_char(local_time, 'YYYYMMDD')=%DATE% )" -o"%FILE%.tmp"
25
26 type %FILE%.tmp > %FILE%
27 del %FILE%.tmp
28 GOTO END
29 :ERROR
30 echo call select_kml.bat veh_id date
31 echo outputs to veh_id.date.kml
32 :END
33 REM ***** Open the file. By Default with Google Earth *****
34 %FILE%
```

## Abbildungsverzeichnis

2.1. Weltweiter Smartphone-Verkauf anhand des Betriebssystems laut Gartner Inc. - 1. Quartal 2010 und 1. Quartal 2011 . . . . .	6
2.2. Smartphone Zufriedenheit . . . . .	8
2.3. Abstraktion der User Experience . . . . .	10
2.4. Typische iOS UI-Elemente anhand der Cityrouter App . . . . .	13
2.5. Screenshot des Apple App Stores (08.08.2011) . . . . .	14
2.6. Sequenz und Phasen von Multi-Touch Gestiken . . . . .	15
2.7. Plattformübergreifende Entwicklung mit Titanium . . . . .	19
2.8. Produktsammlung von Rhomobile . . . . .	20
2.9. Native versus Hybride App . . . . .	22
2.10. Systemaufbau des DLR Taxi-FCD . . . . .	26
2.11. Gesendete Zustände von Taxis und ihre Bedeutung . . . . .	26
2.12. Anzahl von FCD-Fahrzeugen (Taxis) in Großstädten . . . . .	27
2.13. Relativer Fehler im Bezug auf Ausstattungsrate und Intervall . . . . .	30
2.14. Die Basisidee der Positionsbestimmung beim GPS . . . . .	32
2.15. Abstraktion des Systemaufbaus vom Mobilfunknetz . . . . .	35
2.16. GSM Komponenten mit UMTS Luftschnittstellen . . . . .	36
2.17. iOS Abfrage der Ortungserlaubnis . . . . .	39
3.1. Screenshot der DLR Applikation "NAVTEQ-Viewer" . . . . .	43
3.2. Screenshot der Cityrouter Website (04.09.2011 12:30 Uhr) . . . . .	45
3.3. Anforderungsmatrix . . . . .	52
3.4. Bewertungsmatrix . . . . .	54
4.1. Periodische Sendung von FCD als Ablaufplandiagramm . . . . .	61
4.2. Titanium Developer Software (15. August 2011) . . . . .	69

## Abbildungsverzeichnis

---

4.3. Titanium Projekt Struktur . . . . .	71
4.4. Hauptfenster und Tabbar Initialisierung, Auszug der "app.js" . . . .	72
4.5. Event registrieren und ausführen . . . . .	73
4.6. Navigations Button, Auszug von "map_window.js" . . . . .	73
4.7. SOAP Kommunikation, Auszug von "monitor.js" . . . . .	74
4.8. OpenLayers Map mit Touchfunktionalität, Auszug der "openlayers- Mobile.js" . . . . .	75
4.9. Zugriff auf Kartenkacheln, Auszug der "OpenStreetMap.js" . . . . .	76
4.10. Initialisierung der Trafficansicht, Auszug von "openlayersMobile.js" .	76
4.11. Route auf Karte abbilden, Auszug von "openlayersMobile.js" . . . . .	77
4.12. Kompass Zugriff, Auszug von "geolocation.js" . . . . .	78
4.13. Geolokation Bestimmung, Auszug von "geolocation.js" . . . . .	79
4.14. Verifizierung von FCD, Auszug aus "PositionFCD.js" . . . . .	79
5.1. Entwurf der Datenbankstrukturen als Entity Relationship Modell . .	81
5.2. Implementierung eines Datenstruktur-Elements anhand von Hiber- nate und Annotationen . . . . .	83
5.3. Hibernate Konfiguration: Erstellen von Datenbankstrukturen anhand von Java-Klassen, Ausschnitt der Datei "hibernate.cfg.xml" . . . . .	84
5.4. Web Service - die 3 elementaren Bestandteile . . . . .	85
5.5. Web Service: JSON-String zu FCD Objekten formen . . . . .	87
5.6. Web Service: Initialisierung der Message Service und Routing Zugriff, Ausschnitt der Datei "CityrouterService.java" . . . . .	89
5.7. Web Service: Routen Überwachung, Ausschnitt der Datei "Cityrou- terService.java" . . . . .	90
5.8. OSM Karten offline speichern - Ausschnitt von "OpenLayersImage- Loader.java" . . . . .	92
6.1. Ausschnitt einer automatisch erzeugten KML-Datei . . . . .	95
6.2. Intervall von FCD-Sendungen - Ausschnitt von Google Earth . . . .	95
6.3. Farbliche Darstellung der Geoposition anhand der Geschwindigkeit .	96
6.4. FCD-Sammlung - Route: Berlin Friedrichshain nach Teschendorf (Landkreis Oberhavel) 01.10.2011 14:22 - 15:21 Uhr . . . . .	96
6.5. Auswertung des Fragenkataloges (n=2) . . . . .	99

## Abbildungsverzeichnis

---

6.6.	Fehlerhafte Ausführung der App im Android Emulator . . . . .	100
6.7.	Titanium Appcelerator OS feststellen . . . . .	101
6.8.	Ausführung der App im Android Emulator nach Anpassungen . . . .	101
7.1.	Die App in Bildern - Das Karten-Tab . . . . .	104
7.2.	Die App in Bildern - Monitor, Info und Einstellungen . . . . .	104
7.3.	Die App in Bildern - Auswahl einer Route . . . . .	105
B1.	GPS - Sechs Umlaufbahnen mit je vier Satelliten . . . . .	111
B2.	Überblick der Segmente des zivilen (L1) GPS . . . . .	112
B3.	Skizze des Code-Generators . . . . .	114
B4.	Code-Phasen und Chip-Verschiebung . . . . .	115
B5.	Darstellung von GPS Signalerzeugung und Modulation . . . . .	116
B6.	Struktur der Signale und deren Modulation . . . . .	118
B7.	Aufbau eines Datenframes . . . . .	118
B8.	Ausschnitt der Ephemeriden-Daten eines The International GNSS Service (IGS) . . . . .	120
B9.	Ausschnitt der Almanach . . . . .	120

## Literaturverzeichnis

- [Altera Corporation März 2003] ALTERA CORPORATION: *Gold Code Generator Reference Design*, März 2003. – URL [http://www.cs.cmu.edu/~sensing-sensors/readings/Gold\\_Code\\_Generator-an295.pdf](http://www.cs.cmu.edu/~sensing-sensors/readings/Gold_Code_Generator-an295.pdf). – Zugriffsdatum: 18. Juli 2011 - Presented at the ION GNSS Conference 2004
- [Apple Inc. 2010a] APPLE INC.: *CLLocationManager Class Reference*. 2010. – URL [http://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CLLocationManager\\_Class/CLLocationManager\\_Class.pdf](http://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CLLocationManager_Class/CLLocationManager_Class.pdf). – Zugriffsdatum: August 2011
- [Apple Inc. 2010b] APPLE INC.: *iPhone und iPod touch: Grundlegende Informationen zu den Ortungsdiensten*. 2010. – URL [http://support.apple.com/kb/HT1975?viewlocale=de\\_DE](http://support.apple.com/kb/HT1975?viewlocale=de_DE). – Zugriffsdatum: 25. Juli 2011
- [Apple Inc. 2010c] APPLE INC.: *Location Awareness Programming Guide*. 2010. – URL <http://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/LocationAwarenessPG/LocationAwarenessPG.pdf>. – Zugriffsdatum: August 2011
- [Apple Inc. 2011a] APPLE INC.: *Multitouch Events*. 2011. – URL <http://developer.apple.com/library/ios/#DOCUMENTATION/EventHandling/Conceptual/EventHandlingiPhoneOS/MultitouchEvents/MultitouchEvents.html>. – Zugriffsdatum: 6. Juli 2011
- [Apple Inc. 2011b] APPLE INC.: *Technische Daten des iPhone 4*. 2011. – URL <http://www.apple.com/de/iphone/specs.html>. – Zugriffsdatum: 1. August 2011
- [Apple Inc. 23. März 2011] APPLE INC.: *iOS Human Interface Guidelines*. 23. März 2011. – URL <https://developer.apple.com/library/ios/documentation/>

## Literaturverzeichnis

---

- UserExperience/Conceptual/MobileHIG/MobileHIG.pdf. – Zugriffsdatum: 6. Juni 2011
- [Apple Inc. 27. April 2011] APPLE INC.: *Apple Q&A on Location Data*. 27. April 2011. – URL <http://www.apple.com/pr/library/2011/04/27Apple-Q-A-on-Location-Data.html>. – Zugriffsdatum: 25. Juli 2011
- [Apple Inc. Januar 2011] APPLE INC.: *iPhone and iPad: Calibrating Compass*. Januar 2011. – URL <http://support.apple.com/kb/TS2767>. – Zugriffsdatum: 6. Juli 2011
- [Apple Inc. Juni 2010] APPLE INC.: *Privacy Policy*. Juni 2010. – URL <http://www.apple.com/privacy>. – Zugriffsdatum: 25. Juli 2011
- [Barth 25. August 2009] BARTH, Dave: *The bright side of sitting in traffic: Crowd-sourcing road congestion data*. 25. August 2009. – URL <http://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html>. – Zugriffsdatum: September 2011
- [Bitkom 11. Februar 2011] BITKOM: *Jeder fünfte Handynutzer besitzt ein Smartphone*. 11. Februar 2011. – URL [http://www.bitkom.org/files/documents/BITKOM\\_Presseinfo\\_Smartphones\\_14\\_10\\_2010.pdf](http://www.bitkom.org/files/documents/BITKOM_Presseinfo_Smartphones_14_10_2010.pdf). – Zugriffsdatum: 29. Juli 2011
- [Braun Juli 2007] BRAUN, Matthias: *Das GPS-System - Funktionsweise und Einsatzmöglichkeiten im Physikunterricht*. Juli 2007. – URL <http://www.physik.uni-wuerzburg.de/~wilhelm/arbeiten/ZulaGPS.pdf>. – Zugriffsdatum: 4. September 2011
- [Brockfeld 24.05.2011] BROCKFELD, Elmar: *Forschung des DLR zum Verkehrsmanagement / DLR-Institut für Verkehrssystemtechnik - Berlin*. URL [http://www.fav.de/DOCS/tdv11/DLR\\_Brockfeld.pdf](http://www.fav.de/DOCS/tdv11/DLR_Brockfeld.pdf), 24.05.2011. – Forschungsbericht. Zugriffsdatum: 8. Juli 2011
- [Brockfeld u. a. 11–15. Juli 2010] BROCKFELD, Elmar ; SOHR, Alexander ; KRIEG, Sascha: *Quality of Floating Car Data / DLR-Institut für Verkehrssystemtechnik - Berlin*. URL [http://elib.dlr.de/65978/1/FCDQuality\\_WCTR\\_2010\\_final\\_V5.pdf](http://elib.dlr.de/65978/1/FCDQuality_WCTR_2010_final_V5.pdf), 11-15. Juli 2010. – Forschungsbericht

## Literaturverzeichnis

---

- [Brunthaler 2006] BRUNTHALER, Stefan: *Requirements Analysis & Management*. Vorlesungsskript. 2006. – URL [http://www.tfh-wildau.de/sbruntha/Material/SDGM/VL\\_SDGM\\_Requirements.pdf](http://www.tfh-wildau.de/sbruntha/Material/SDGM/VL_SDGM_Requirements.pdf). – Zugriffsdatum: 5. August 2011
- [Cheng 15. Juli 2011] CHENG, Roger: ITC says HTC violating two of Apple's patents. In: *Cnet News* (15. Juli 2011). – URL [http://news.cnet.com/8301-1035\\_3-20079905-94/itc-says-htc-violating-two-of-apples-patents](http://news.cnet.com/8301-1035_3-20079905-94/itc-says-htc-violating-two-of-apples-patents). – Zugriffsdatum: 7. August 2011
- [Dam 2010] DAM, Eliot: *Digital Map Technology Has Arrived*. 2010. – URL <http://www.cicomra.org.ar/cicomra2/expocomm/TUTORIAL%208%20Dam-%20NAVTEQ.pdf>. – Zugriffsdatum: August 2011
- [Deng und Li 12–14 Dez. 2008] DENG, Zhong-hua ; LI, Qin: Upgrading the User Experience of Digital Library. In: *International Conference on Computer Science and Software Engineering on 2008*. IEEE Computer Society on IEEE/IET Electronic Library (IEL), VDE VERLAG, 12-14 Dez. 2008, S. 832–835. – URL <http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=4723032>. – Zugriffsdatum: 1. August 2011. – ISBN 978-0-7695-3336-0
- [Dodel und Häupler 2009a] DODEL, Hans ; HÄUPLER, Dieter: *Satellitennavigation*. Springer Verlag, 2009. – ISBN 978-3540794431
- [Dodel und Häupler 2009b] DODEL, Hans ; HÄUPLER, Dieter: *Satellitennavigation*. Springer Verlag, 2009. – ISBN 978-3540794431
- [Douangboupoupha 2009] DOUANGBOUPHA, Phavanhna: US Smart Phone Platforms / R2integrated. URL [http://www.r2integrated.com/Portals/2/White-Papers/R2i\\_WP\\_SmartPhoneComparison.pdf](http://www.r2integrated.com/Portals/2/White-Papers/R2i_WP_SmartPhoneComparison.pdf), 2009. – Forschungsbericht. Zugriffsdatum: 3. Juli 2011
- [El-Rabbany 2002] EL-RABBANY, Ahmed: *Introduction to GPS: the Global Positioning System*. Artech House, 2002. – ISBN 1-58053-183-1
- [Elektronik Kompendium ] Elektronik Kompendium (Veranst.): *GPS - Global Positioning System*. – URL <http://www.elektronik-kompendium.de/sites/kom/1201071.htm>. – Zugriffsdatum: 11. Juli 2011



## Literaturverzeichnis

---

- [European Commission 2011] EUROPEAN COMMISSION, Enterprise&Industry: *Satellite navigation*. 2011. – URL [http://ec.europa.eu/enterprise/policies/satnav/galileo/index\\_en.htm](http://ec.europa.eu/enterprise/policies/satnav/galileo/index_en.htm). – Zugriffsdatum: 2. September 2011
- [Europäischen Union 28 Juni 2004] EUROPÄISCHEN UNION, Pressemitteilung: *GALILEO und GPS navigieren Seit' an Seit': EU und USA unterzeichnen abschließendes Abkommen*. 28 Juni 2004. – URL <http://europa.eu/rapid/pressReleasesAction.do?reference=IP/04/805&format=PDF>. – Zugriffsdatum: 2. September 2011
- [Eylert 2005] EYLERT, Bernd: *The Mobile Multimedia Business: Requirements and Solutions*. John Wiley & Sons, 2005. – ISBN 978-0470012345
- [Eylert 3. Auflage 2010] EYLERT, Bernd: *Kompendium Numerische Mathematik*. Verlag News & Media, Berlin, 3. Auflage 2010. – ISBN 978-3936527094
- [Eylert WS 2007] EYLERT, Bernd: *Core-Network Elements - Mobilkommunikation*. WS 2007. – Vorlesung im Studienfach Mobilkommunikation an der TH Wildau (Wintersemester)
- [Fabig 1995] FABIG, Anselm: *Konzept eines digitalen Empfängers für die Funknavigation mit optimierten Algorithmen zur Signaldemodulation*, Technischen Universität Berlin, Dissertation, 1995. – URL <http://www.navtec.de/personen/acf/dd/dd.pdf>. – Zugriffsdatum: 19. Juli 2011
- [Friedrich 31. Oktober 2007] FRIEDRICH, Markus: Nutzung von Mobilfunkdaten für das Verkehrsmanagement - Das Projekt Do-It / Universität Stuttgart. URL [http://www.mvi.baden-wuerttemberg.de/servlet/is/66408/13\\_Das\\_Projekt\\_Do-iT.pdf](http://www.mvi.baden-wuerttemberg.de/servlet/is/66408/13_Das_Projekt_Do-iT.pdf), 31. Oktober 2007. – Forschungsbericht. Zugriffsdatum: 25. Juli 2011
- [Gist 2007] GIST, Chris: *Global Positioning System*. 2007. – URL <http://www.lib.virginia.edu/scholarslab/resources/class/intro2GIS/introToGPS.pdf>. – GIS Specialist, Zugriff: Oktober 2011
- [Gold Oktober 1967] GOLD, Robert: *Optimal binary sequences for spread spectrum multiplexing*. Oktober 1967. – URL <http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=5681968>. – Zugriffsdatum: 3. Oktober 2011

## Literaturverzeichnis

---

- [GPSWorld 4. April 2011] GPSWORLD: *Russia's MTS Now Selling Smartphone Supporting GLONASS*. 4. April 2011. – URL <http://www.gpsworld.com/gnss-system/glonass/news/russia-s-mts-now-selling-smartphone-supporting-glonass-11464>. – Zugriffsdatum: 2. September 2011
- [Gutmann Juni 2007] GUTMANN, Frank: *Positionsbestimmung in GSM- und UMTS-Netzwerken*. Juni 2007. – Seminararbeit
- [IEEE-Standards 2010] IEEE-STANDARDS, Association: *Standard Group MAC Addresses: A Tutorial Guide*. 2010. – URL [standards.ieee.org/develop/regauth/tut/macgrp.pdf](http://standards.ieee.org/develop/regauth/tut/macgrp.pdf). – Zugriffsdatum: 25. Juli 2011
- [IEEE1233 1998] IEEE1233: IEEE Guide for Developing System Requirements Specifications / The Institute of Electrical and Electronics Engineers, Inc. URL <http://ieeexplore.ieee.org/iel4/5982/16016/00741940.pdf>, 1998. – Forschungsbericht. Zugriffsdatum: 5. August 2011
- [IEEE802.11 2007] IEEE802.11: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. 2007. – URL <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>. – IEEE Computer Society, Zugriff: Oktober 2011
- [Jordans Oktober 2008] JORDANS, Tobias: Donald Norman definiert User Experience / nexum AG. URL <http://uxzentrisch.de/donald-norman-definiert-user-experience/>, Oktober 2008. – Forschungsbericht. Zugriffsdatum: 1. August 2011
- [JSON ] JSON: *Introducing JSON*. – URL <http://www.json.org/>. – Zugriffsdatum: 25. Mai 2011
- [Kaplan und Hegarty 2006] KAPLAN, Elliott ; HEGARTY, Christopher: *Understanding GPS: principles and applications*. Artech House, 2006. – ISBN 978-1580538947
- [Karunanayake u. a. 2004] KARUNANAYAKE, M.D. ; CANNON, M.E. ; LACHAPPELLE, G.: Evaluation of Assisted GPS (AGPS) in Weak Signal Environments

## Literaturverzeichnis

---

- Using a Hardware Simulator / Department of Geomatics Engineering - University of Calgary, Canada. URL [http://plan.geomatics.ucalgary.ca/papers/04gnss\\_ion\\_dharetal.pdf](http://plan.geomatics.ucalgary.ca/papers/04gnss_ion_dharetal.pdf), 2004. – Forschungsbericht. Zugriffsdatum: 18. Juli 2011 - Presented at the ION GNSS Conference 2004
- [Kerkau Mai 2011] KERKAU, Florian: Mobile Monitor 2011 / Goldmedia. URL <http://www.goldmedia.com/publikationen/studien/mobile-interac/studie-mobile-monitor-2011.html>, Mai 2011. – Forschungsbericht. Zugriffsdatum: 29. Juli 2011
- [Käfert 2008] KÄFERT, Robert: *Entwicklung eines Konzeptes für eine mobile Applikation für das Apple iPhone*, TFH Wildau, Diplomarbeit, 2008
- [Köhne und Wöckner 2006] KÖHNE, Anja ; WÖSSNER, Michael: *GPS Satellite Orbits*. 2006. – URL <http://www.kowoma.de/gps/Satelliten.htm>. – Zugriffsdatum: 11. Juli 2011
- [Köhne und Wöckner 2007] KÖHNE, Anja ; WÖSSNER, Michael: *Positionsbestimmung*. 2007. – URL <http://www.kowoma.de/gps/Positionsbestimmung.htm>. – Zugriffsdatum: 11. Juli 2011
- [Köhne und Wöckner 2009] KÖHNE, Anja ; WÖSSNER, Michael: *GPS Satellite Orbits*. 2009. – URL <http://www.kowoma.de/en/gps/orbits.htm>. – Zugriffsdatum: 11. Juli 2011
- [Langley 1991] LANGLEY, Richard: *GPS Satellite Orbits* / University of New Brunswick. URL <http://gauss.gge.unb.ca/gpsworld/EarlyInnovationColumns/Innov.1991.03.pdf>, 1991. – Forschungsbericht. Zugriffsdatum: 11. Juli 2011
- [Levinson April 2011] LEVINSON, Alex: *3 Major Issues with the Latest iPhone Tracking "Discovery"*. April 2011. – URL <https://alexlevinson.wordpress.com/2011/04/21/3-major-issues-with-the-latest-iphone-tracking-discovery/>. – Zugriffsdatum: 25. Juli 2011
- [Levinson u. a. 2011] LEVINSON, Alex (Hrsg.) ; STACKPOLE, Bill (Hrsg.) ; JOHNSON, Daryl (Hrsg.): *Third Party Application Forensics on Apple Mobile Devices*. 2011. (44th Hawaii International Conference on System Sciences). – ISBN 978-0-7695-4282-9

## Literaturverzeichnis

---

- [Lin und Ye 16–17. Mai 2009] LIN, Feida ; YE, Weiguo: Operating System Battle in the Ecosystem of Smartphone Industry. In: MUHIN, V.E. (Hrsg.) ; YE, Zhiwei (Hrsg.): *The First International Symposium on Information Engineering and Electronic Commerce - IEEC 2009*. IEEE Computer Society on IEEE/IET Electronic Library (IEL), VDE VERLAG, 16-17. Mai 2009, S. 617–621. – URL <http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=5175193>. – Zugriffsdatum: 3. Juli 2011. – ISBN 978-0-7695-3686-6
- [Mansfeld 2009] MANSFELD, Werner: *Satellitenortung und Navigation: Grundlagen, Wirkungsweise und Anwendung globaler Satellitennavigationssysteme*. Vieweg+Teubner Verlag, 2009. – ISBN 978-3834806116
- [Mark und LaMarche 2009] MARK, Dave ; LAMARCHE, Jeff: *Beginning iPhone Development*. Apress, 2009. – ISBN 978-1430216261
- [Mattsson 2001] MATTSSON, Oskar: *Positioning of a cellular phone using the SIM*, Royal Institute of Technology Kungliga Tekniska Högskolan, Diplomarbeit, 2001
- [Müller 11. Februar 2011] MÜLLER, Peter: Nokia und Microsoft schließen Mobilfunkallianz / IDG Magazine Media GmbH. URL [http://www.macwelt.de/artikel/\\_News/375736/nokia\\_und\\_microsoft\\_schliessen\\_mobilfunkallianz/1](http://www.macwelt.de/artikel/_News/375736/nokia_und_microsoft_schliessen_mobilfunkallianz/1), 11. Februar 2011. – Forschungsbericht. Zugriffsdatum: 29. Juli 2011
- [Nitobi 2011] NITOBİ: *Story*. PhoneGap. 2011. – URL <http://www.phonegap.com/about/story>. – Zugriffsdatum: 10. August 2011
- [Nokia 11. Februar 2011 – London] NOKIA: *Nokia and Microsoft announce plans for a broad strategic partnership to build a new global ecosystem*. 11. Februar 2011 - London. – URL <http://press.nokia.com/2011/02/11/>. – Zugriffsdatum: 29. Juli 2011
- [Ong 16. Juli 2011] ONG, Josh: ITC ruling against HTC may spell trouble for other Android makers. In: *Apple Insider* (16. Juli 2011). – URL [http://www.appleinsider.com/articles/11/07/16/itc\\_ruling\\_against\\_htc\\_may\\_spell\\_trouble\\_for\\_other\\_android\\_makers.html](http://www.appleinsider.com/articles/11/07/16/itc_ruling_against_htc_may_spell_trouble_for_other_android_makers.html). – Zugriffsdatum: 7. August 2011

## Literaturverzeichnis

---

- [OpenLayers 2011] OPENLAYERS: *Spherical Mercator*. 2011. – URL [http://docs.openlayers.org/library/spherical\\_mercator.html](http://docs.openlayers.org/library/spherical_mercator.html). – Zugriffsdatum: 2. September 2011
- [Paeger 2007] PAEGER, Jürgen: *Der große Luftocean - die Atmosphäre*. 2007. – URL <http://www.oekosystem-erde.de/html/luft.html>. – Zugriffsdatum: 11. Juli 2011
- [Pettery und Goasduff 26. Januar 2011] PETTEY, Christy ; GOASDUFF, Laurence: Gartner Says Worldwide Mobile Application Store Revenue Forecast to Surpass \$15 Billion in 2011 / Gartner Inc. URL <http://www.gartner.com/it/page.jsp?id=1529214>, 26. Januar 2011. – Forschungsbericht. Zugriffsdatum: 25. Mai 2011
- [Pettery und Stevens 19. Mai 2011] PETTEY, Christy ; STEVENS, Holly: Gartner Says 428 Million Mobile Communication Devices Sold Worldwide in First Quarter 2011, a 19 Percent Increase Year-on-Year / Gartner Inc. URL <http://www.gartner.com/it/page.jsp?id=1689814>, 19. Mai 2011. – Forschungsbericht. Zugriffsdatum: 25. Mai 2011
- [Pettery und Stevens 7. April 2011] PETTEY, Christy ; STEVENS, Holly: Gartner Says Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year-End 2012 / Gartner Inc. URL <http://www.gartner.com/it/page.jsp?id=1622614>, 7. April 2011. – Forschungsbericht. Zugriffsdatum: 29. Juli 2011
- [Pocketnavigation 21. April 2011] POCKETNAVIGATION: *NAVIGON traffic4all informiert über die aktuelle Verkehrslage*. 21. April 2011. – URL [http://www.pocketnavigation.de/news/view\\_2617\\_\\_navigon-traffic4all-informiert-ueber-die-aktuelle-verkehrslage/1.1.88.html](http://www.pocketnavigation.de/news/view_2617__navigon-traffic4all-informiert-ueber-die-aktuelle-verkehrslage/1.1.88.html). – Zugriffsdatum: August 2011
- [Presstext.Redaktion 14. Juli 2011] PRESSETEXT.REDAKTION: *Google Maps liefert nun Echtzeit-Verkehrsinfos*. 14. Juli 2011. – URL <http://search.presstext.com/news/20110714025>. – Zugriffsdatum: September 2011
- [Prinz 2009] PRINZ, Torsten: *Auswertungsverfahren*. 2009. – URL [http://ivvgeo.uni-muenster.de/Vorlesung/GPS\\_Script/messung\\_verfahren.html](http://ivvgeo.uni-muenster.de/Vorlesung/GPS_Script/messung_verfahren.html). – Zugriffsdatum: 28. Juli 2011

## Literaturverzeichnis

---

- [Rhomobile 2011] RHOMOBILE: *Positionsbestimmung in GSM- und UMTS-Netzwerken*. 2011. – URL <http://rhomobile.com/products/rhodes/>. – Zugriffsdatum: 25. Juli 2011
- [Rodewig 2009] RODEWIG, Klaus M.: *Apps entwickeln für iPhone und iPad*. Galileo Press, 2009. – ISBN 978-3836214636
- [Sacher 12. Mai 2000] SACHER, Marc: Cäsium-Atomuhr / Universität Bielefeld - Fachbereich Physik. URL <http://www.physik.uni-bielefeld.de/~msacher/scripte/csuhr.pdf>, 12. Mai 2000. – Forschungsbericht
- [Schmidt 17. Juli 2009] SCHMIDT, Timo: *Umgang und Optimierungsmöglichkeiten datenintensiver Prozesse in Service-orientierten Architekturen(SOA)*. Bachelorarbeit, TH Wildau. 17. Juli 2009. – URL <http://elib.dlr.de/60226/>. – Zugriffsdatum: 7. August 2011
- [Sohr u. a. 2010a] SOHR, Alexander ; BROCKFELD, Elmar ; EBENDT, Rüdiger: Validation of a Taxi-FCD system / Institut für Verkehrssystemtechnik des DLR e.V. URL <http://elib.dlr.de/66060/1/TP014-4.pdf>, 2010. – Forschungsbericht. Zugriffsdatum: 7. August 2011
- [Sohr und Wagner 2008] SOHR, Alexander ; WAGNER, Peter: Short term traffic prediction using cluster analysis based on floating car data / Institut für Verkehrssystemtechnik des DLR e.V. URL [http://elib.dlr.de/57318/1/Sohr\\_ShortTermPrediction\\_abstract-ITS-2008.pdf](http://elib.dlr.de/57318/1/Sohr_ShortTermPrediction_abstract-ITS-2008.pdf), 2008. – Forschungsbericht. Zugriffsdatum: 7. August 2011
- [Sohr u. a. 2010b] SOHR, Alexander ; WAGNER, Peter ; KUHNS, Günter ; EBENDT, Rüdiger ; BROCKFELD, Elmar: Self Evaluation of Floating Car Data Based on Travel Times from Actual Vehicle Trajectories / Institut für Verkehrssystemtechnik des DLR e.V. URL [http://elib.dlr.de/70336/1/FCD\\_Self\\_Evaluation\\_final.pdf](http://elib.dlr.de/70336/1/FCD_Self_Evaluation_final.pdf), 2010. – Forschungsbericht. Zugriffsdatum: 7. August 2011
- [Song und Sun 22–25 Nov. 2008] SONG, Yu ; SUN, Shouqian: Research on Virtual Exhibition Interaction and User Experience Technology. 9th International Conference on CAID/CD 2008. In: *Computer-Aided Industrial Design and Conceptual Design*. IEEE/IET Electronic Library (IEL), VDE VERLAG, 22-25 Nov.

## Literaturverzeichnis

---

- 2008, S. 1360–1363. – URL <http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=4730816>. – Zugriffsdatum: 1. August 2011. – ISBN 978-1-4244-3290-5
- [Spiering und Haiges 14. Juni 2010] SPIERING, Markus ; HAIGES, Sven: *HTML5-Apps für iPhone und Android*. Franzis Verlag, 14. Juni 2010. – ISBN 978-3645600521
- [Stäuble 2009] STÄUBLE, Markus: *Programmieren fürs iPhone*. dpunkt Verlag, 2009. – ISBN 978-3898646352
- [Teker November 2005] TEKER, Ugur: *Realisierung und Evaluation eines Indoor-Lokalisierungssystems mittels WLAN*, Universität Bremen Fachbereich Informatik, Diplomarbeit, November 2005. – URL [http://www.sfbtr8.spatial-cognition.de/project/r1/theses/diplomarbeit\\_ugur\\_teker.pdf](http://www.sfbtr8.spatial-cognition.de/project/r1/theses/diplomarbeit_ugur_teker.pdf). – Zugriffsdatum: 25. Juli 2011
- [Thomanek 4. Mai] THOMANEK, Rico: *Grundlagen der iOS-Programmierung*. 4. Mai. – URL [http://www.staff.hs-mittweida.de/~rthomane/grdl\\_iphone/Grdl\\_iPhone\\_Programmierung.pdf](http://www.staff.hs-mittweida.de/~rthomane/grdl_iphone/Grdl_iPhone_Programmierung.pdf). – Zugriffsdatum: 25. Mai 2011
- [Titanium 2011] TITANIUM, Appcelerator: *Titanium makes cross-platform native application development easy*. 2011. – URL <http://www.appcelerator.com/products/titanium-cross-platform-application-development/>. – Zugriffsdatum: 10. August 2011
- [Titanium Juni 2009] TITANIUM, Appcelerator: *Appcelerator Titanium Empowers Web Developers to Quickly Develop Native iPhone and Android Applications*. Juni 2009. – URL <http://www.appcelerator.com/2009/06/titanium-beta/>. – Zugriffsdatum: 10. August 2011, Press Release
- [Ullenboom 2009] ULLENBOOM, Christian: *Java ist auch eine Insel*. Galileo Press, 2009. – ISBN 978-3836213714
- [USCG ] USCG: *General information on GPS*. – URL <http://www.navcen.uscg.gov/?pageName=GPS>. – Zugriffsdatum: 18. Juli 2011
- [USCG 2. Juni 1995] USCG: *Global Positioning System Standard Positioning Service Signal Specification*, 2. Juni 1995. – URL <http://www.navcen.uscg.gov/pubs/gps/sigspec/gpsps1.pdf>. – Zugriffsdatum: 19. Juli 2011

## Literaturverzeichnis

---

- [USPTO Juni 2011] USPTO: *Portable multifunction device, method, and graphical user interface for translating displayed content*. Juni 2011. – URL <http://patft.uspto.gov/netacgi/nph-Parser?patentnumber=7966578>. – Zugriffsdatum: 1. August 2011
- [Veckenstedt 2006] VECKENSTEDT, Maik: *Die Ionosphäre*. 2006. – URL [http://www.pci.tu-bs.de/aggericke/PC5-Atmos/Poster/Ionosph%E4re\\_M\\_Veckenstedt.pdf](http://www.pci.tu-bs.de/aggericke/PC5-Atmos/Poster/Ionosph%E4re_M_Veckenstedt.pdf). – Zugriffsdatum: 11. Juli 2011
- [Voigt 2009] VOIGT, Gabriel: *Identifikation und Visualisierung von gestauten Verkehrsmustern in Straßennetzen mit Hilfe von FCD*, TFH Wildau, Diplomarbeit, 2009
- [von Fuchs Dezember 2008] VON FUCHS, Teresa: Open Source RIA Platform. In: *CrunchBase* (Dezember 2008). – URL <http://techcrunch.com/2008/12/09/appcelerator-raises-41-million-for-open-source-ria-platform/>. – Zugriffsdatum: 7. August 2011
- [W3C 2002] W3C: *Web Services Activity Statement*. 2002. – URL <http://www.w3.org/2002/ws/Activity>. – Zugriffsdatum: 1. Oktober 2011
- [Wagner 2005] WAGNER, Stefan: *Entwicklung und Implementierung eines Algorithmus zur Generierung von Verkehrsmeldungen für den motorisierten Straßenverkehr, in urban Gebieten, basierend auf stark heterogenen Reisezeitinformationen*, TFH Wildau, Diplomarbeit, 2005
- [Winkler 2009] WINKLER, Lutz: *WLAN - Wireless Local Area Network*. 2009. – URL <http://www.staff.hs-mittweida.de/~win/vorlesungen/wlan.pdf>. – Zugriffsdatum: 25. Juli 2011
- [Zhang 17–19. Nov. 2010] ZHANG, Dongyu: PDA Design based on Embedded Linux. In: *11th International Conference on Computer-Aided Industrial Design & Conceptual Design (CAIDCD) - IEEE 2010*. IEEE/IET Electronic Library (IEL), VDE VERLAG, 17–19. Nov. 2010, S. 1299–1302. – URL <http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=5681968>. – Zugriffsdatum: 3. Juli 2011. – ISBN 978-1-4244-7973-3



## Literaturverzeichnis

---

[Zogg 2002] ZOGG, Jean-Marie: *GPS Basics*. 2002. – URL [www.inovatrack.com/gps/GPS\\_basics\\_u\\_blox\\_en.pdf](http://www.inovatrack.com/gps/GPS_basics_u_blox_en.pdf). – Zugriffsdatum: 18. Juli 2011